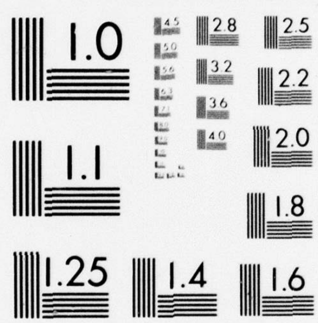
AD-A061 539 PURDUE UNIV LAFAYETTE IND SCHOOL OF ELECTRICAL ENGI--ETC F/G 9/4
CODING OF AERIAL RECONNAISSANCE IMAGES FOR TRANSMISSION OVER NO--ETC(U)
SEP 78 O R MITCHELL, S C BASS, E J DELP F30602-75-C-0082

UNCLASSIFIED RADC-TR-78-210 NL

1 OF 2
AD
A061539



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A061539

LEVEL II

12

18 19

RADC-TR-78-210
Final Technical Report
September 1978

6

CODING OF AERIAL RECONNAISSANCE IMAGES FOR TRANSMISSION OVER NOISY CHANNELS.

10 O. R. Mitchell,
S. C. Bass,
E. J. Delp
T. W. Goeddel

Purdue University

15 F30602-75-C-0082

11 Sep 78

12 134p.

9 Final technical rept.
1 Apr 77-1 May 78,

DDC FILE COPY

Laboratory Directors' Fund No. 017112PO

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

292 000 JOB

78 11 20 004

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-78-210 has been reviewed and is approved for publication.

APPROVED: John T. Gamble

JOHN T. GAMBLE
Project Engineer

APPROVED: Fred I. Diamond

FRED I. DIAMOND
Technical Director
Communications and Control Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

This effort was funded totally by the Laboratory Directors' Fund.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-78-210	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CODING OF AERIAL RECONNAISSANCE IMAGES FOR TRANSMISSION OVER NOISY CHANNELS		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 1 Apr 77 - 1 May 78
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) O. R. Mitchell T. W. Goeddel S. C. Bass E. J. Delp		8. CONTRACT OR GRANT NUMBER(s) F30602-75-C-0082
9. PERFORMING ORGANIZATION NAME AND ADDRESS Purdue University School of Electrical Engineering W. Lafayette IN 47907		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61101F 017112P0
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (DCCL) Griffiss AFB NY 13441		12. REPORT DATE September 1978
		13. NUMBER OF PAGES 134
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Enginer: John T. Gamble (DCCL)

This effort was funded totally by the Laboratory Directors' Fund.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Coding
Image coding
Aerial reconnaissance
Photo analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Several picture coding techniques were applied to aerial reconnaissance photos for the purpose of comparative evaluation. Both spatial and frequency domain methods were used: five of them previously known, as well as two new ones enunciated here for the first time.

The comparisons reported within were subjective ones carried out by a team of five professional photo analysts trained to extract "targets" from aerial images. (see reverse)

DD FORM 1 JAN 73 1473 next page

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

78 11 20 004

20 (Cont'd)

Their judgements were obtained on noise-free images as well as those coded and subjected to a simulated noisy transmission channel prior to decoding. The bit error rates used were 0, 10^{-4}, 10^{-3}, and 10^{-2}.
.0001 .001 .01

The specific coding methods evaluated were: 1) fixed zonal, 3) threshold, 3) four-class zonal (Chen and Smith), 4) hybrid, 5) Micro-Adaptive Picture Sequencing (CDC); and the two new techniques: 6) block truncation, and 7) variable zone. Detailed theoretical discussions of each coding method are included as are listings of all specific parameter choices used in each method. Arithmetic operations count estimates and the resulting mean-square error performance for each technique are also given.

Acknowledgements

We gratefully acknowledge the support, helpful criticism, and sugestions provided by Dr. John Gamble of RADC. His grasp of the field of practical image coding made him an important contributor to our effort. We also thank E. E. Hicks and Robert LaSalle, both also of RADC, for their helpful discussions.

Very special thanks are due Richard R. Petroski and his colleagues of Rome Research Corporation who provided the expert photo evaluations included in this report.

Additional research assistants involved in this project were Steve Carlton, Po Chen, Paul Stiling, and Ali Tabatabai. This effort could not have been accomplished without their long hours, dedication, and patience.

Table of Contents

LIST OF FIGURES

LIST OF FIGURES (Cont.)

LIST OF FIGURES (Cont.)

LIST OF FIGURES (Cont.)

LIST OF TABLES

EVALUATION

This effort performed an objective comparison of the effect of source/channel encoding in transmitting digitized, high resolution imagery. The bases of this comparison were mean squared error and, more significantly, the assessment of professional photo-interpreter/analysts (the ultimate users of such imagery). Source encoding is intended to reduce non-essential (to the user) redundancy in original imagery, but this increases the sensitivity of the source-encoded data to communication channel noise. Thus channel encoding (efficient bit apportionment/quantization, data formatting, and added redundancy for error detection and correction) is necessary to minimize this noise sensitivity. The present research effort, in assessing the performance of such combined coding strategies under realistic noisy channel conditions, provides a more credible basis for reducing the bandwidth requirements for intelligence-quality imagery communication.

John T. Gamble
JOHN T. GAMBLE
Project Engineer

I. Introduction

This project arose from the need to evaluate various coding methods for transmitting digital imagery. Images to be considered were black and white aerial photographs and the ultimate quality triterion was the human image analyst's evaluation. The ultimate goal is to find how to achieve data compression, image fidelity, and noise immunity simultaneously.

The following guidelines were used in our research:

(1) Promising techniques were to be implemented and evaluated using aerial imagery and experienced analysts.

(2) Redundancy in both picture directions were to be exploited in the coding.

(3) Hybrid combinations of methods were encouraged.

(4) The effects of the human visual system were to be considered.

(5) The effects of channel errors on the reconstructed image were to be demonstrated.

(6) System complexity was to be considered.

(7) Only single frame, black and white imagery was to be used.

I.1 Summary of Research Effort

This report summarizes the results of a one-year effort on this task by Purdue University. To achieve a good comparison, transform, hybrid, and spatial coding techniques were implemented on our PDP 11/45 computer. After preliminary screening, some techniques were modified or eliminated, and new coding methods were developed. The best methods were then used to code three original pictures at various channel error rates. The resulting pictures have been rated by qualified photo-analysts. All results are presented and summarized in Section III of this report.

I.2. Discussion of Coding Methods Investigated

This section presents an overview of the coding methods investigated during our research. A detailed discussion of each method used in our final comparisons is presented in Section II of this report.

I.2.1. Two-Dimensional Transform Techniques

A recent performance comparison of coding methods [1] has listed the two-dimensional transform technique as the best performing method. We selected several transforms for comparison based on published performance. These were the Fourier [2], Cosine [3], Slant [4], Haar [5], and Walsh [6]. Basis function images for the last four are shown in Figs. I-1 to I-4. As discussed in Section II, the Cosine Transform was found to be superior within this group as far as mean square error and subjective quality was concerned. Therefore, the Cosine Transform was used as the basic two-dimensional transform method within the several transform techniques.

The block size to be used for the two-dimensional transform was also investigated. It was found that working with 8x8 pixel blocks did not allow the compression possible with 16x16 blocks and no significant improvement was found for larger block sizes. Therefore, the final transform implementation was done over 16x16 blocks.

There were four two-dimensional transform techniques used in the final comparison:

(1) Fixed zone with variable bit assignment;

(2) Adaptive zone controlled by the percent energy content within the zone;

(3) Four class zone with the classes determined by total a.c. energy in each block; and

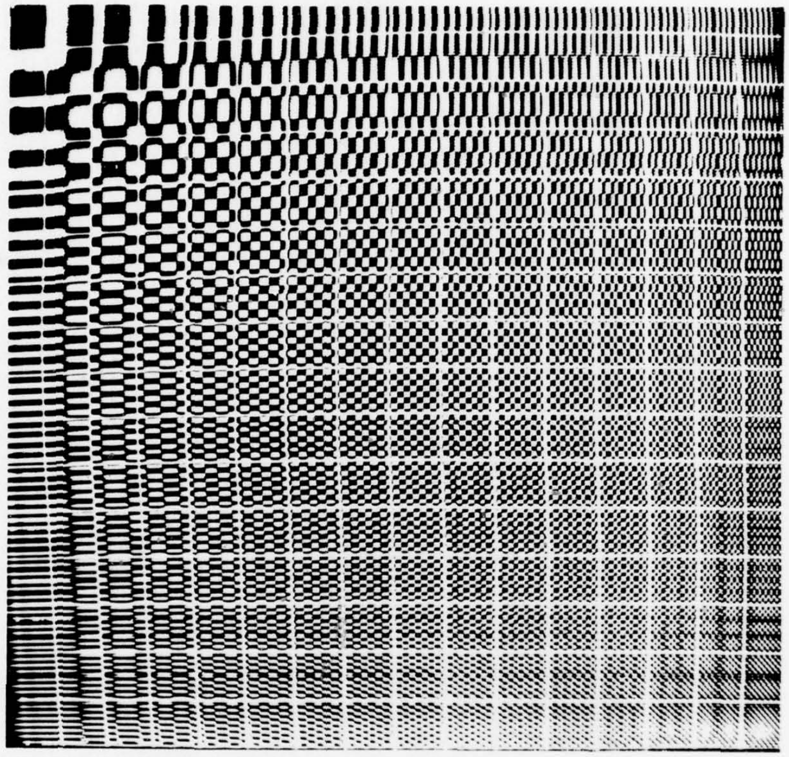


Figure I-1. The 256 basis function pictures for the two-dimensional 16x16 Cosine Transform.

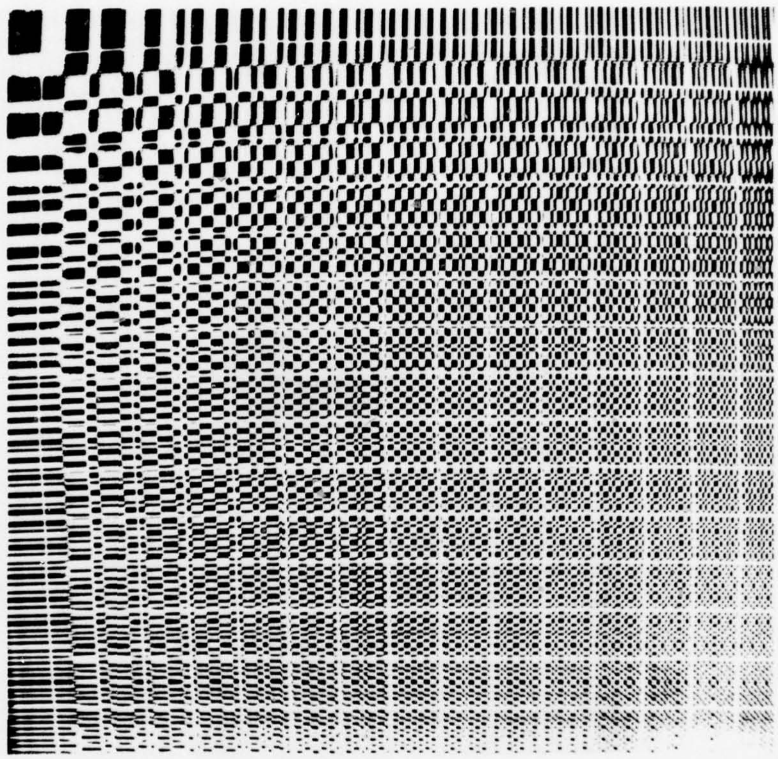


Figure I-2. The 256 basis function pictures for the two-dimensional 16x16 Slant Transform.

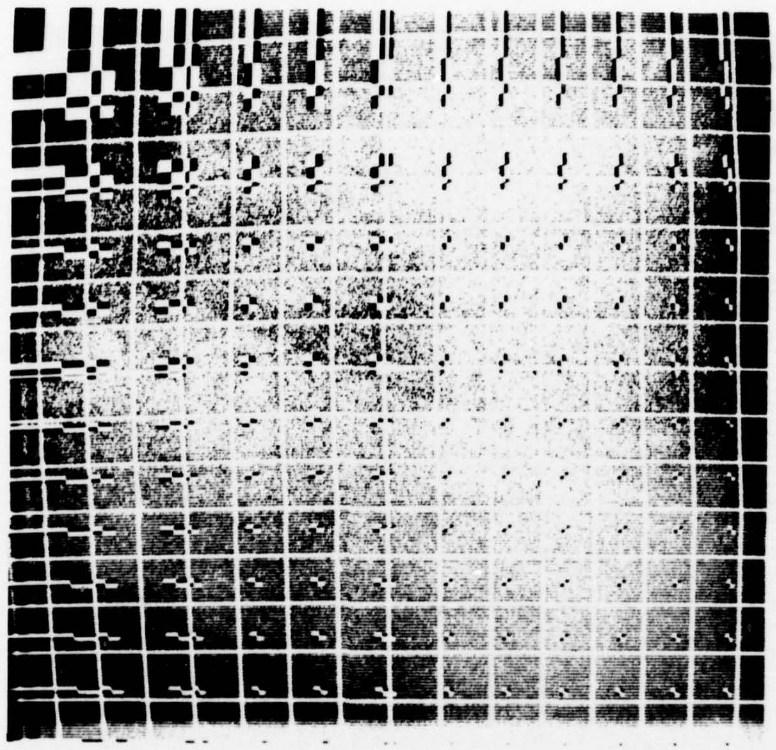


Figure I-3. The 256 basis function pictures for the two-dimensional 16x16 Haar Transform.

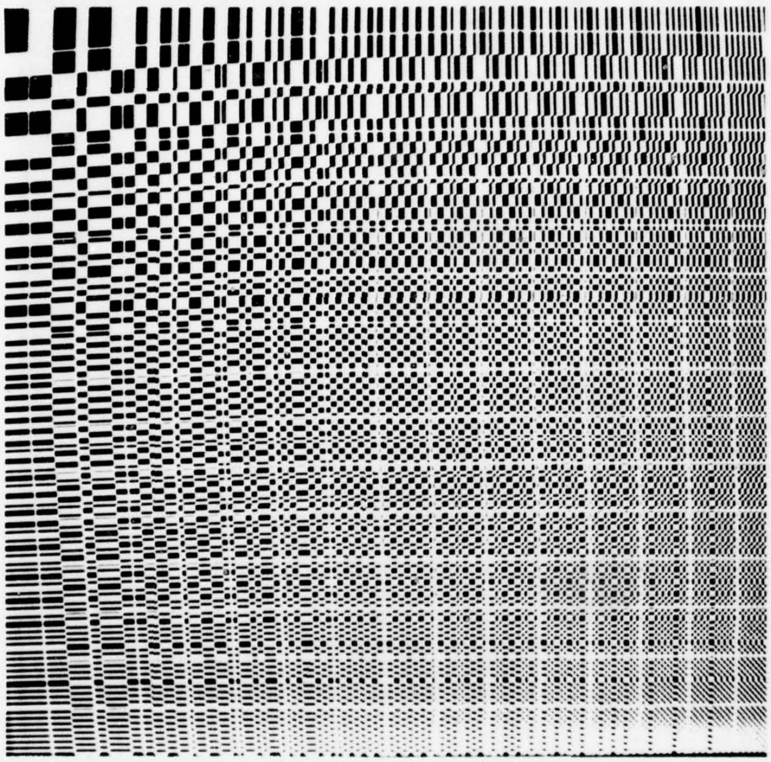


Figure I-4. The 256 basis function pictures for the two-dimensional 16x16 Walsh Transform.

(4) Threshold of coefficients in each block.

The zone methods are well known for their noise immunity but tend to cause some blurring in the reconstruction. The threshhold method gives some improvement at the cost of more noise sensitivity and reduced coding efficiency. The transform methods are the most computationally intensive of the investigated methods.

I.2.2. Hybrid Technique

The hybrid technique was introduced by Habibi [7] to keep the advantages of transform coding while reducing the required computation. In our implementation, a one-dimensional cosine transform was taken across 32 sample segments of the rows and DPCM coding was used to code vertically the Cosine Transform coefficients. Variable bit assignment was used for the difference coefficients. Updating of the data using PCM was periodically done to prevent propagation of channel errors.

I.2.3. Spatial Techniques

In order to eliminate even more computation and possibly to provide a new environment for developing coding improvements, several spatial (non-transform) techniques were investigated. These were predictive, adaptive sequencing, and block truncation.

I.2.3.1. Predictive Methods

The most prevelant method of spatial coding is to predict the next picture value from past values and quantize only the "error" signal, which is defined as the difference between the actual and predicted values [8,34]. The predictive methods have not demonstrated the compression ability of the transform methods and are also much more sensitive to channel errors [9, 1]. We did implement predictive methods but the compression achieved for high fidelity was not comparable to the other spatial methods. These predictive techniques are described in section II.4.3.

I.2.3.2. Micro Adaptive Picture Sequencing (MAPS)

This method has recently been developed by Control Data Corporation for RADC [10]. Pixels within a spatial region are combined and replaced with a single value. The size of the region is highly adaptive so that areas with large changes are kept at high resolution while low detail regions are highly compressed. This technique is computationally simple and can be implemented using no multiplies in the algorithm.

I.2.3.3. Block Truncation

This method has been recently developed at Purdue [11]. Each 4x4 subblock of the image is quantized into two levels such that, on reconstruction, the local mean and variance is preserved. This method has demonstrated a good match to the human visual system and is very simple computationally.

I.3. Other Considerations

Many directions were pursued in initally developing the final comparisons. Some of these considerations are now described.

I.3.1. Human Visual System Response

The psychophysical properties of the human visual system [12, 25] are often of use in designing coding systems. Image information not visible to a human observer can be deleted prior to coding and the remaining information can be weighted such that the most visible information is coded using a larger number of bits. However, the unique nature of our evaluation procedure presents a problem: Human analysts may well examine the image at several magnifications. The parts of an image not visible at one position may become visible as the analyst zooms in on that particular region. Therefore, we did not attempt to directly use psychophysical properties in our coding schemes.

It should be noted that some coding conditions (such as pilot TV displays) result in fixed observation situations and the human visual system psychophysics can then be used to great advantage.

I.3.2. Subjective and Objective Fidelity Criteria

Most coding studies have used the mean square error between the original and reconstructed picture as the basis for quality assessment. It is well known that certain types of distortion are much more objectionable than others even though they result in the same mean square error. In general, noise correlated with the picture, or noise having structure (edges, lines), is much more detrimental than uncorrelated random noise [25].

Therefore, although mean square error was measured for the techniques tested, the ultimate criterion was the subjective ranking of pictures by experienced photo-analysts.

I.3.3. Dependence of Results on Data Used

It is often hard to compare various coding techniques when each technique was used to code a different set of data. We have found a large difference in the response of certain coding methods to a "head and shoulders" picture (high correlation between pixels) and an aerial photograph (when an airport runway may be only one or two pixels wide). This project has concentrated on the latter type data.

The presentation of a coding result at a given compression rate in bits/pixel is useless unless the number of pixels involved is also given. It is also impossible to subjectively judge the performance of coding methods when the results are shown with the pixels so densely packed that the effects of the coding on individual pixels is not visible. This is especially true when the analyst/observer will be examining reconstructed images with a magnifying lens. We, therefore, have displayed our images for comparison in this report using 512x512 images at 7 1/2" square. This results in less than 70 pixels per inch, which allows for careful comparison of the various techniques.

I.3.4. Channel Errors

It was necessary to consider the effects of channel errors for the codes tested since the implementation of the coding is to be on a low bandwidth, error-prone communication link. Only random binary symmetric channel errors were considered (burst errors can be transformed to this type by data interleaving) with error probabilities of 10^{-4}, 10^{-3}, and 10^{-2}.

In some of the coding schemes, some of the overhead bits were very important to preserving the integrity of the reconstructed code. These bits should, of course, be highly error protected. In order to conserve programming time and effort, these overhead bits were not subjected to the simulated error channel; however, the number of overhead bits required was multiplied by 3 in the compression calculations to make allowance for an error correcting code to be used. This is also the method used by Chen and Smith

[13] in determining error performance. Section II includes reference to this error control simulation under each method which used it and describes which overhead information was so treated.

II. Theoretical Discussion of Coding Methods

A thorough description of the seven coding methods used for the final subjective tests plus one additional method now follows. These eight techniques included four two-dimensional transform methods, a hybrid technique, and three spatial methods. The DPCM spatial technique was not used on the final data set dicussed in Section III.

II.1. Some Channel Coding Tools

The following coding method discussions reference the use of variable bit assignment and non-uniform quantization. These two tools will first be described prior to discussing the separate coding techniques.

II.1.1. Variable Bit Assignment

Although transform coefficients are usually relatively uncorrelated, they do vary considerably in amplitude with the low frequency coefficients having the largest probability of being larger. In order to optimize the quantization of such a set of coefficients, it is necessary to assign more quantization levels to the coefficients with the largest variance. This is consistent with results from rate distortion theory [14].

First let PS, BS denote respectively the picture size and block size along one dimension of a square (e.g., PS = 512, BS = 16). Bits are assigned to each transform coefficient on the basis of the sample variance of the coefficients within each class (more than one class occurs only for the four class zone method described in Section II.2.3); in particular, we first compute the sample variance of the coefficients as follows:

$$\sigma_k^2(u,v) = \frac{K(BS)^2}{PS^2} \sum_{m \epsilon k} \hat{f}_m^2(u,v) \tag{1}$$

$$(u,v)\neq 0,\ u<16,\ v<16$$
$$m=1,\ldots(\frac{PS}{BS})^2$$
$$k=1,\ldots,K$$

where

$\hat{f}_m^2(u,v)$ (u,v)th element of mth block

$\sigma_k^2(u,v)$ ensemble average of the variance of (u,v)th element belonging to kth class.

K number of classes (K=1 or K=4 for all of our methods).

Then bits are assigned to each coefficient according to the following formula:

$$NB_k(u,v) = \frac{1}{2} Log_2 \sigma_k^2(u,v) - Log_2 D \tag{II-1}$$

where

$NB_k(u,v)$ = number of bits assigned to (u,v)th element belonging to kth class.

D = constant number

The parameter D can be determined using the following steps:

Let

MAX = number of bit assigned to each dc coefficient

AVG desired average bits/pixel

Then we can write:

$$\sum_{k} \sum_{(u,v)\neq 0} NB_k(u,v) + (K)(MAX) = (AVG)(K)(BS)^2$$

$$\sum_{K} \sum_{(u,v)\neq 0} Log_2 \frac{\sigma_k(u,v)}{D} + K(MAX) = AVG(K)(BS)^2$$

$$\sum_{K} \sum_{(u,v)\neq 0} Log_2\sigma_k(u,v) - K(BS^2-1)Log_2 D + K(MAX) = AVG(K)(BS)^2$$

$$K(BS^2-1)Log_2 D = \sum_{k} \sum_{(u,v)\neq 0} Log_2\sigma_k(u,v) + K(MAX) - AVG(K)(BS)^2$$

$$Log_2 D = \frac{\sum_{k} \sum_{(u,v)\neq 0} Log_2\sigma_k^2(u,v)}{2K(BS^2-1)} + \frac{MAX}{BS^2-1} - \frac{BS^2 AVG}{(BS^2-1)} \qquad (II-2)$$

Notice that MAX bits are assigned automatically to the dc component.

Now because of the round-off error, we may not achieve the desired number of bits in the first trial. Therefore, $Log_2 D$ can be modified as:

$$\sum_{k} \sum_{(u,v)\neq 0} NB'_k(u,v) - \sum_{k} \sum_{(u,v)\neq 0} NB_k(u,v) = TB - DB \qquad (II-3)$$

where

$NB'_K(U,V)$ denotes values assigned in the first trial

TB Total hits assigned during the first trial

DB desired number of bits

it follows from equation (II-3) that:

$$\sum_k \sum_{(u,v)\neq 0} \left(\mathrm{Log}_2 \frac{\sigma_k(u,v)}{D'} - \mathrm{Log}_2 \frac{\sigma_k(u,v)}{D}\right) = TB - DB$$

$$\sum_k \sum_{(u,v)\neq 0} \mathrm{Log}_2 \frac{D}{D'} = TB - DB$$

$$K(BS^2-1)\mathrm{Log}_2 \frac{D}{D'} = TB - DB$$

$$K(BS^2-1)\mathrm{Log}_2 D = K(BS^2-1)\mathrm{Log}_2 D' + (TB - DB)$$

$$\mathrm{Log}_2 D = \mathrm{Log}_2 D' + \frac{TB - DB}{K(BS^2-1)} \qquad \text{(II-4)}$$

II.1.2. Non-Uniform Quantization

When the probability density function of the samples to be quantized is not uniform, the mean-square error due to quantization can be reduced by using non-uniform spacing between successive quantum levels. For the a.c. transform coefficients, a zero mean Gaussian random variable was assumed. For the hybrid technique, the DCPM produces a Laplacian (two-sided exponential) density function.

The details of the optimum quantizer have been developed by Max [15]. We used a table look-up procedure to implement the quantization. The coefficients were normalized by their standard deviations and the bit assignment for a given coefficient determined which quantization table was used for assigning the code word. Again, these tables are derived by Max [15].

II.2. Two-Dimensional Transform Techniques

The two-dimensional "transform" of an NxN array of discrete-space data is simply the representation of that array as a finite weighted sum of "basis functions." Each basis function (or basis picture) is itself a discrete-space array, and the "transform" of the original data is taken to mean the <u>collection of weights</u> applied to these functions in order to match their (weighted) sum to the original array. For all transform types in use, the number of distinct NxN basis functions required to represent any NxN "original" is N^2.

As an example, the discrete-space cosine transform basis functions are

$$\cos\left[\frac{\pi(2m+1)j}{2N}\right]\cos\left[\frac{\pi(2n+1)k}{2N}\right] \qquad \text{(II-4)}$$

for j, k, m, and n independently ranging from 0 through N-1. The symbols m and n are spatial coordinates that index the picture elements (pixels) from the upper lefthand corner (of the NxN array) to the lower right. A set of cosine basis pictures are shown in Fig. I-1. Since the sampled cosine basis functions are also N^2 in number, the weights associated with them can also be arranged in an NxN array, called the transform array (or, often, simply "transform"). The j and k coordinates establish the basis function horizontal and vertical spatial frequencies, and can also be used to label any basis function in the cosine set <u>or</u> its weight. The latter (that of indices referencing transform weights) is most common. Here the upper lefthand corner of an array of transform weights ($j=k=0$) is taken to be the coefficient applied to the "d.c." basis function. As we work our way to the lower righthand corner of the transform array ($j=k=N-1$), we obtain the weight multiplying the cosine function of highest horizontal <u>and</u> vertical spatial frequency. In the case of the cosine transform, all transform coefficients are real scalars.

Other transforms, (i.e., basis function sets) in use are the Fourier, Haar, Slant, Kachunen-Loeve (KL), and the various Walsh transforms. All of these are discussed in [16]. The KL transform is recognized to provide the best mean-square error fit* by tailoring its basis functions to the statistical properties of the data set at hand. It produces transform weights that are fully uncorrelated [16]. However, KL is rarely used as no computationally efficient algorithm for its calculation is known, <u>and</u> it requires an exact model of the image covariance function.

The Fourier and KL methods can result in N^2 complex weighting coefficients, with half of these being trivially derivable from the remaining ones. The other four transforms are "real" in that they always produce real coefficients when applied to real data arrays.

The transform of a picture is rarely taken as a single operation over the entire MxM scene. Rather the picture is usually first broken into numerous square contiguous "blocks" of much smaller size, say with $N \ll M$. A reason for this is that a "full scene" transform would require far more computational labor to produce. As an example, the two dimensional Fourier transform of an NxN pixel array requires an operation count proportional to:

---

*When fewer than a full set of N^2 basis functions are used.

$$N^2 \log_2 N$$

where an "operation" is defined as a multiplication-addition pair. If $M = 512$ and $N = 16$, then two applications of this formula demonstrate that the ratio of the two operation counts is

$$\frac{\log_2 M}{\log_2 N} = 2.25$$

so that "blocking" results in a 56% savings in computational labor. More important, however, are the memory/disk transfer considerations. At eight bits per pixel, storage of an entire 512x512 picture within a computer's memory would require roughly 1/4 million bytes of memory storage. Since most transforms require that the <u>full</u> scene be referenced with the computation of <u>every</u> transform weight, having less memory available than required necessitates a large amount of auxiliary memory (disk, etc.) swapping. Such core-disk swapping can easily slow down a transform computation by an order of magnitude or more. From a theoretical point of view, the gross non-homogeneous nature of large images also makes it necessary to use small sub-images.

As mentioned in Section I.2.1, a block size of 16x16 was found adequate to give almost optimum compression for the cosine block transform. This is consistent with the mean-square error performance versus block size presented in [1].

Suppose that a picture to undergo transmission has been subdivided into square, contiguous blocks of size 16x16. The transform of this 256-element array may be taken. At this point, one has the choice of transmitting either the original spatial block (using PCM coding, predictive coding, or other spatial techniques) <u>or</u> the array of transform coefficients. In either case, there are 256 scalars to be transmitted, and each can be satisfactorily represented with an average of eight binary bits (256 quantization levels). Thus no bandwidth compression advantage is held by either representation.

In order to reduce the number of information bits to be passed through a transmission channel, one may choose to send only a subset of the total number of information quantities required to represent the 16x16 block perfectly. If transform weights (samples) are to be transmitted, a number of basis function coefficients may be deleted from the transmission message with the usual understanding being that the channel receiver will set all missing coefficients to zero. The result is a reconstructed block (at the receiver) that is somewhat in error with respect to the original. A property of most (but not all) transforms is that this error will be distributed over the entire reconstructed block, rather than being concentrated within any single region.

Any algorithm providing a decision of which transform samples to transmit, and which to ignore, can be called a transform sample reduction technique. Clearly the mean-square error over a reconstructed block will be minimized if the neglected samples are small or expected to be small. Over a large number of blocks, one invariably finds that those transform coefficients near d.c. are frequently large, while those associated with the higher spatial frequencies are typically small. Thus a simple-minded sample reduction algorithm might single out a square or circular zone of "transmittable" coefficients, all clustered about the d.c. weight. (see Fig. II-1). This is the most basic variation of the technique known as "zonal" sample reduction.

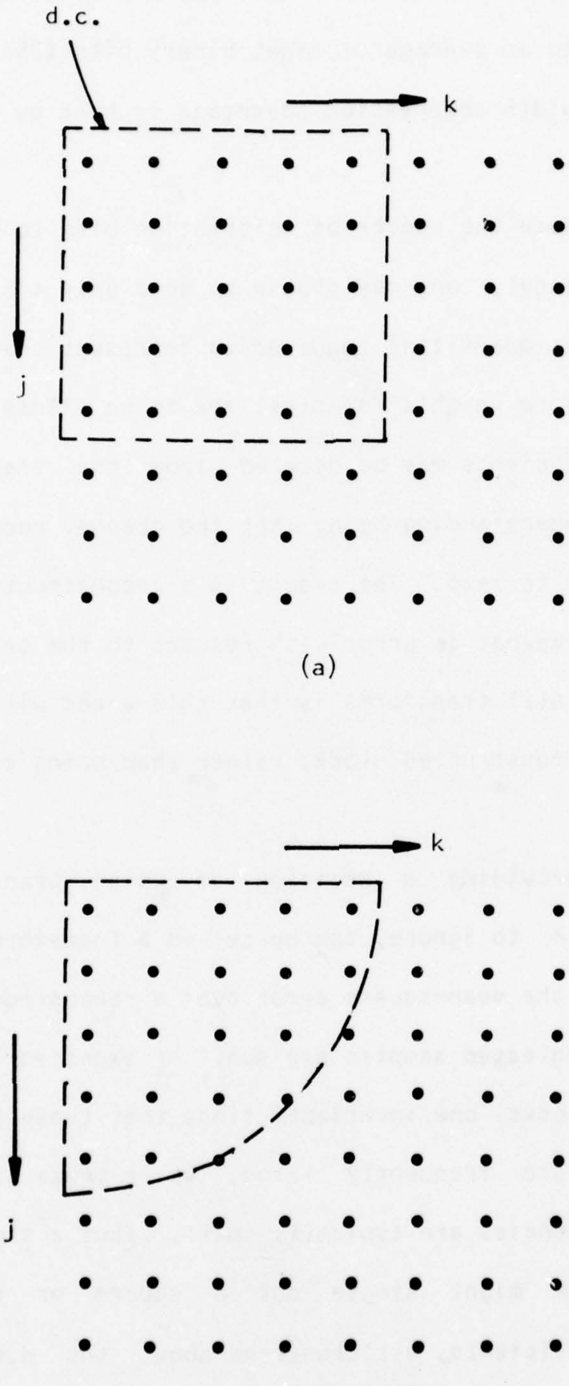


Fig. II-1: Two possible "zone" subsets of transform coefficients to be transmitted over a restricted bandwidth channel: (a) square and (b) circular.

8	5	4	4	3	3	2	2	2	2	1	1	1	1	1	1
6	4	4	3	3	3	2	2	2	2	1	1	1	1	1	1
5	4	4	3	3	3	2	2	2	1	1	1	1	1	1	1
4	3	3	3	3	3	2	2	2	1	1	1	1	1	1	1
4	3	3	3	3	2	2	2	2	1	1	1	1	1	1	1
3	3	3	3	3	2	2	2	2	1	1	1	1	1	1	1
3	3	3	3	2	2	2	2	1	1	1	1	1	1	0	0
3	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
2	2	2	2	2	2	1	1	1	1	1	1	1	1	0	0
3	2	2	2	2	2	2	1	1	1	1	1	1	1	1	0
3	2	2	2	1	1	1	1	1	1	1	1	1	0	0	0
2	2	2	1	1	1	1	1	1	1	1	1	0	0	0	0
3	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0
3	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

Table II-1 Bit Assignment Table for Fixed Zone on AIRPORT 2 at 1.5 bits/pixel.

Of the four transform techniques implemented for final comparison, three are zonal in nature. The fixed zone method used the variable bit assignment to establish the zone used for the entire picture; the adaptive zone method allowed the fixed zone to be adaptively modified based on the percent a.c. energy within the zone and the four class zone method breaks the blocks into four categories based on total a.c. energy, and then uses the variable bit assignment to set four fixed zones.

The last transform technique (threshhold) selects the largest coefficients within each block regardless of position and transmits those values.

II.2.1. Fixed Zone Block Transform

This method was our simplest block (two-dimensional) transform method. One pass was made over the picture to find the variance of each a.c. cosine transform array coefficient across every 16x16 block in the picture. The bit assignment algorithm described in Section II.1.1 is then used to apportion the quantization bits among the transform coefficients. For example, to achieve an average compression rate of 1.5 bits/pixel, each 16x16 transform array would have to be represented with 256x1.5 = 384 bits (this neglects the overhead discussed later). The d.c. term is always quantized with 8 bits and was never modified by the variable bit assignment. This leaves 376 bits to be distributed to the a.c. coefficients. An example is shown in Table II-1. This is the bit assignment for the original picture AIRPORT2 (see Fig. III-3) for a compression rate of 1.5 bits/pixel.

Note that the assignment of zero bits to various coefficients has automatically applied a sample reduction to the set of all transform coefficients: Those coefficients given zero transmission bits simply won't be transmitted. Note that a zone arrived at in this way need not be connected.

Any fixed zone method has an advantage in transmission over noisy channels in that each block will be transmitted in a fixed, known number of bits. Thus, it is impossible for the erroneous toggling of any information bit to result in the receiver losing synchronization with the transmitter (which often results in the garbling of the remainder of the picture). However, prior to actual picture transmission, the receiver must be informed of the number of information bits that the transmitter has reserved for each transform coefficient. The zone method results in Section III included the transmission of 256 numbers each ranging from zero to eight and indicating the number of information bits to be assumed by the receiver for the associated transform weight.

Though two coefficients may be assigned an identical number of information bits, it is not necessary that the analog values assigned to their quantization levels also be identical. Even if the probability density functions associated with these coefficients are of the same functional form (e.g., Gaussian), their respective quantization level values are chosen in accord with their respective variances, and these will rarely be identical. The aim of fitting quantum levels to coefficient statistics is, of course, to minimize the quantization error of that coefficient over the entire set of picture blocks. This was done by normalizing each coefficient by its standard deviation and using an optimum quantization as described in Section II.1.2.

The fixed zonal method tested during this effort used the nonlinear quantizers of [15] for coefficients assigned lengths five bits or less. Linear quantizers (also derived in [15]) were applied to quantities of lengths 6-8 bits. Thus, the spacing between successive quantum levels in any coefficient is simply a known factor (listed in [15]) <u>times</u> the calculated standard deviation. As in [13], our program uses a single coefficient variance (namely the largest variance calculated amongst all one-bit coefficients) upon which to base quantizer scale factors for all transmitted weights: If V is the largest "one-bit" variance, then the variance assumed for all coefficients of bit length n is $V2^{n-1}$. In order to inform the receiver of this basic scale factor, our method reserves 32 bits for the transmission of its square root (the standard deviation). An insignificant improvement in overall compression could be obtained by quantizing this standard deviation to fewer bits.

An undetected error in a received standard deviation will cause a (possibly large) error to appear in all the values a certain coefficient location takes on throughout a picture. Thus, in the presence of channel noise, some error correcting capability must be allowed for in the coding of this information. Even worse would be an undetected error in a coefficient bit length number. Upon receipt of the first such error coefficient, the receiver will lose transmitter synchronization.

Thus the scale factor and bit assignment table should be error protected. We treated these two items as discussed in Section I.3.4 to allow for error protection down to error probabilities of 10^{-2}. The block diagram and approximate number of calculations required for a 512x512 picture for this method are shown in Fig. II-2.

II.2.2. Adaptive Zone Block Transform

As discussed in Section II.2.1, the zonal sample reduction method tested in this work responds to transform coefficient statistics as computed (and averaged) over an entire picture. Thus in an overall sense, this technique adapts itself to any given scene (and compression rate). However, it

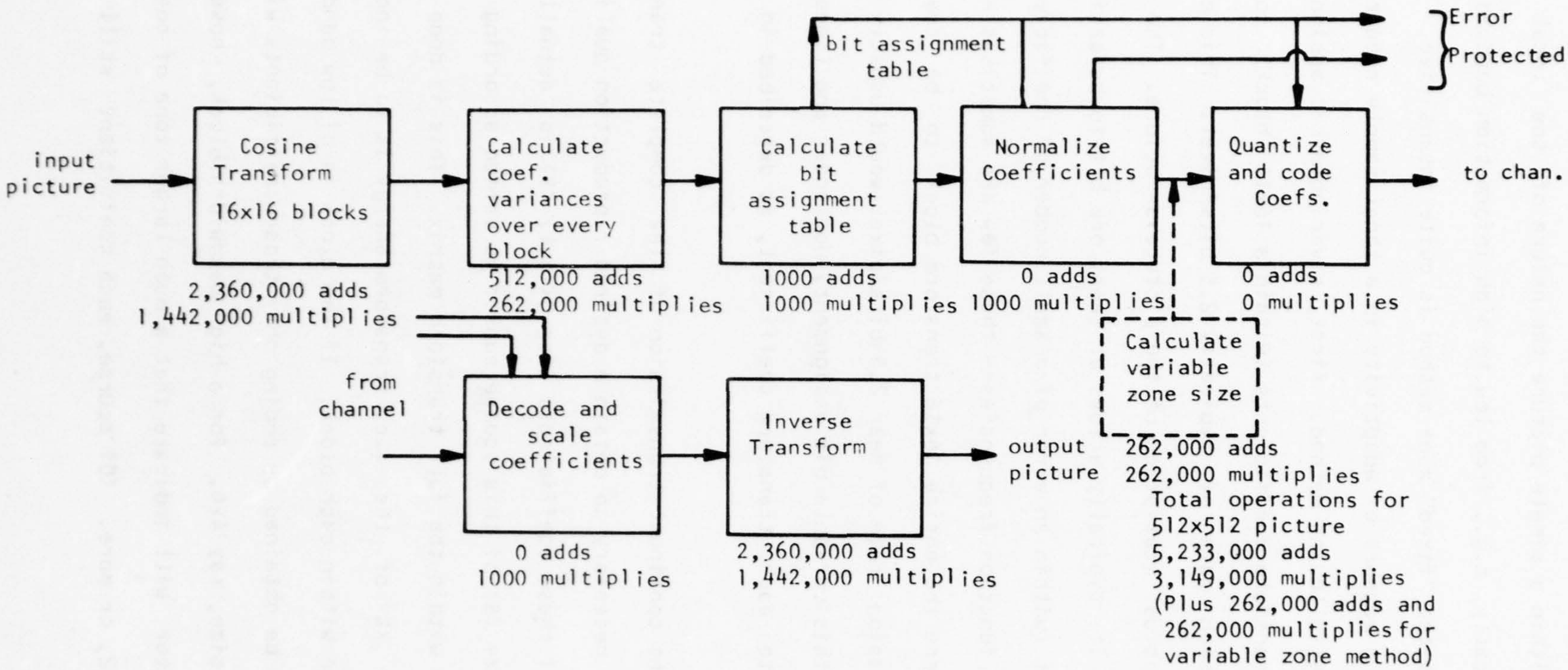


Fig. II-2. Fixed Zone Block Transform Coding. Additional labor needed by "variable zone" method is shown in dashed enclosure.

is clear that within a single picture the nature of the local information can change radically, e.g., from low to high information bandwidth. To such local variations the "fixed" zone method is quite insensitive.

To provide a measure of adaptivity in a zonal sample reduction context, we proposed the following method. First, a variable bit assignment is made to all cosine transform coefficients (within a 16x16 block) to achieve a basic transmission rate of typically 2.5 bits/pixel. This assignment is done using the actual statistics of the picture as before. The value of 2.5 is common as it typically allows at least one bit to be assigned to each transform weight (within an error of a small number of coefficients at the highest basis function frequencies-- these few are sometimes assigned zero bits). Thus, were the entire 16x16 transform block to be transmitted, a minimum compression rate of near 2.5 bits/pixel would usually be achieved. In addition to this variable bit assignment, nonlinear and linear quantizers were applied to each transform coefficient, as described in the previous section.

However, the continual transmission of the complete transform array will rarely be necessary to obtain a desired reproduction quality. Thus, a square subset of these coefficients (for $0 \leq (j,k) \leq S$) is actually transmitted, and the size (S) of this square subset is varied according to the energy distribution within the full transform matrix. This is done by prescribing that only X% of the a.c. transform energy is to be included in the transmitted zone within each block. If the block is of low bandwidths, this percentage can be obtained by coding only those coefficients within a small square zone of size, say 4x4. For a high bandwidth block, however, the X percentage factor will indicate that a much larger zone of coefficients be sent: say 12x12, or more. (Of course, each coefficient still occupies a number of representation bits assigned to it by the initial bit assignment algorithm.)

The result is a method that retains much of the error "hardness" of a fixed zone approach, and yet can adapt to local information bandwidth changes. In practice the method has been observed to adjust adaptively the square zone size over a range from 1x1 to 15x15 (at 1.5 bits/pixel).

Of course the square zone size, which is a number ranging from one to 15, must be passed to the receiver at the head of each packet of coefficients. During the tests discussed below, these zone size bits, the bit length values, and the "basic" coefficient standard deviation (used to scale each coefficient quantizer) were transmitted using error protection as discussed in Section I.3.4.

It is often necessary in communication applications to be able to prescribe the bits/pixel compression rate B to result from a given coding algorithm. In the case of the adaptive zone method, this implies that B must be transcribable into a percent-energy figure X. We have devised the following technique (which remains to be tested). Let $\overline{E}(S)$ denote the average percent-energy figure resulting when a constant square zone size of SxS is used over all picture blocks. More precisely,

$$\overline{E}(S) = \frac{1}{N^2}\sum_{I=0}^{N-1}\sum_{J=0}^{N-1}\left[\frac{\sum\limits_{i=0}^{S-1}\sum\limits_{j=0}^{S-1}(c_{ij}^{IJ})^2}{\sum\limits_{i=0}^{15}\sum\limits_{j=0}^{15}(c_{ij}^{IJ})^2}\right] X100\%, \quad i+j \neq 0$$

Here the picture is taken to be of dimensions 16Nx16N. also, c_{ij}^{IJ} is the transform weight associated with the (i,j) basis function as used to represent block (I,J). Note that the squares of these weights are already available, as they are also needed by the variable bit assignment algorithm.

Once the bit assignment algorithm has been applied, to a given picture, it is then an easy task to construct the function N(S) which indicates the bits/pixel compression that would result if a constant square zone size of SxS were used for all blocks.

Note that both $\overline{E}(S)$ and N(S) are monotonically increasing functions of S. Further, $\overline{E}(16) = 100\%$, and N(16) is the full block size coding compression. As mentioned above, this latter is typically 2.5 bits/pixel.

In all but the most unusual pictures, N(S) will not only be monotonic but strictly so. Hence, N is invertible, and given a prescribed compression rate, B, $S = N^{-1}(B)$ will produce a square zone size that if used uniformly over all blocks would result in B bits/pixel compression. Then, $\overline{E}(S)$ will return an average percent energy figure that, again, would result from a constant square zone size SxS. We will set $X = \overline{E}(S)$ to validate this technique in our future tests.

II.2.3. Four Class Zone Block Transform

This method is identical to that described by Chen and Smith in [13]. In order to allow some adaptivity over a picture, one of four possible zones is used for each block. Each 16x16 block is classified into one of four equal size classes based on the total variance of that block (or the transform a.c. energy). An example is shown in Table II-2 for the picture AIRPORT2 (Fig. III-3). This requires one pass through the data to collect a histogram of the energy in each block. On the second pass, the cosine transform coefficient variances for each class are calculated. This results in four 16x16 variance matrices.

The variable bit assignment (see Section II.1.1) is then used to assign bits to all coefficients over all classes simultaneously. An example is

3	3	2	1	4	3	2	2	2	3	1	4	4	2	3	2	2	2	3	4	1	3	4	2	2	3	2	3	2	2	3	4
1	1	1	1	2	2	2	3	3	4	2	2	3	4	1	2	4	4	1	4	2	1	1	4	1	3	4	2	4	1	2	4
1	1	4	1	1	2	2	1	3	3	2	3	3	1	1	3	3	1	2	4	3	2	1	4	2	1	2	3	3	3	3	4
1	4	1	1	1	3	1	1	4	2	1	3	3	1	3	2	2	2	2	3	1	1	1	4	4	1	2	2	2	3	3	3
3	4	1	4	3	1	4	2	4	1	4	2	3	2	1	1	1	2	2	3	3	2	1	1	2	1	1	3	1	2	4	3
1	1	2	2	1	2	3	3	1	1	3	3	3	4	2	2	1	2	3	2	3	2	2	1	2	2	2	1	2	4	4	2
1	1	2	1	1	2	2	2	1	2	3	3	3	4	2	2	3	1	1	2	4	4	2	1	2	1	1	1	1	2	1	3
1	4	4	1	2	1	2	3	2	1	1	3	1	2	2	2	1	4	4	4	2	3	2	4	1	2	3	4	2	1	2	1
1	1	1	3	2	3	2	4	4	1	3	1	3	3	2	2	1	1	2	1	1	3	1	2	3	3	4	4	2	3	1	2
4	2	2	2	2	1	2	1	4	2	2	3	1	2	1	2	2	2	4	2	2	2	4	3	4	3	4	2	4	4	3	4
2	3	4	2	3	1	1	2	2	1	2	2	2	1	1	2	1	3	4	3	3	4	4	4	4	2	4	3	1	3	3	2
2	2	2	1	1	1	1	1	2	3	4	1	2	4	1	1	1	3	3	3	3	3	2	4	2	2	3	4	1	1	2	3
2	2	4	1	4	1	1	2	1	1	2	2	1	3	3	2	1	1	2	3	4	3	4	1	2	2	2	2	3	1	2	3
2	4	4	1	2	1	1	3	3	4	1	1	1	2	2	1	1	1	3	2	4	3	3	1	2	2	3	1	3	4	4	4
2	4	3	1	2	1	4	2	1	2	1	2	2	1	1	1	1	1	4	1	4	4	1	1	2	1	2	2	4	2	3	3
1	3	1	2	3	1	3	1	3	4	2	2	1	3	1	1	1	1	1	2	3	4	1	1	3	2	1	3	4	3	4	4
3	3	2	2	1	1	1	1	1	4	3	4	1	4	1	3	2	1	1	1	2	4	2	1	1	4	2	4	4	4	4	4
2	2	2	3	3	3	1	2	3	4	4	2	3	3	1	4	3	2	2	2	2	4	3	1	2	3	3	3	2	4	3	3
2	1	1	1	1	4	1	2	3	1	1	1	3	3	2	2	1	1	4	4	3	3	4	4	3	3	3	4	4	3	2	3
4	1	3	4	1	1	2	1	3	4	1	3	1	2	1	2	2	1	2	4	4	4	4	4	4	3	4	4	3	3	4	4
2	2	2	1	1	4	3	1	3	4	4	1	1	1	1	3	2	1	2	4	4	4	2	3	4	4	3	3	3	4	3	4
4	4	1	1	1	2	4	1	1	1	2	4	2	1	1	2	1	1	3	4	3	4	3	4	3	3	4	4	4	3	3	3
1	4	2	1	1	1	1	2	4	1	1	1	3	2	3	3	4	4	4	4	4	4	4	4	2	4	4	4	3	3	4	3
3	4	1	2	1	1	1	2	1	2	1	1	2	2	4	2	4	3	4	4	4	4	3	3	3	4	4	4	3	3	3	3
2	1	2	1	2	4	3	2	2	2	3	1	4	3	3	2	4	3	3	3	3	2	2	3	3	4	4	4	3	4	3	4
1	1	1	1	2	4	2	1	3	2	1	3	4	3	2	2	3	2	4	2	3	2	3	3	4	4	4	4	4	4	4	3
4	2	1	4	1	4	4	2	2	4	2	1	2	4	4	3	3	2	3	4	3	4	3	3	4	4	4	4	2	4	3	3
2	1	1	2	4	1	1	2	3	2	3	1	2	3	3	3	4	3	4	3	4	3	3	3	3	4	3	2	4	4	3	2
1	1	4	1	1	1	1	1	4	4	4	4	2	4	3	3	2	3	2	3	4	3	3	3	3	4	3	4	4	4	3	2
3	2	1	3	1	2	2	1	4	4	2	2	2	3	3	3	3	4	3	4	4	3	2	2	4	4	4	4	4	4	2	2
3	1	4	4	3	2	1	1	3	2	2	4	4	4	4	3	2	3	3	4	4	2	2	3	4	3	4	4	3	3	3	3
2	3	4	4	1	3	1	1	3	3	3	4	4	4	4	3	3	4	4	4	4	3	3	4	4	4	4	4	3	4	4	3

Table II-2. Energy classification map for each 16x16 block on AIRPORT2. (Fig. III-3). Class 1 has the highest a.c. energy.

Class 1

8	6	5	4	4	3	3	3	2	2	2	2	1	1	1	1
6	5	4	4	3	3	3	2	2	2	2	2	1	1	1	1
5	4	4	4	3	3	3	2	2	2	2	2	1	1	1	1
4	4	4	3	3	3	3	2	2	2	1	1	1	1	1	1
4	4	3	3	3	3	3	2	2	2	2	1	1	1	1	1
3	3	3	3	3	3	2	2	2	2	1	1	1	1	1	1
3	3	3	3	3	3	2	2	2	2	1	1	1	1	1	1
3	2	3	2	2	2	2	2	2	2	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1
3	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
2	2	2	1	2	1	1	1	1	1	1	1	1	1	0	1
2	2	1	1	1	1	1	1	1	1	1	1	1	0	1	0
2	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0
2	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
2	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

Class 2

8	4	4	4	3	3	2	2	2	2	2	1	1	1	1	1
6	4	4	3	3	3	2	2	2	2	2	1	1	1	1	1
5	4	4	3	3	3	3	2	2	2	1	1	1	1	1	1
4	4	3	3	3	3	2	2	2	2	1	1	1	1	1	1
4	3	3	3	3	3	2	2	2	2	1	1	1	1	1	1
3	3	3	3	3	3	2	2	2	1	1	1	1	1	1	1
4	3	3	3	2	2	2	2	2	1	1	1	1	1	1	1
3	3	3	2	2	2	2	2	2	1	1	1	1	1	1	0
2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1
3	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1
3	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
2	2	2	1	2	1	1	1	1	1	1	1	1	0	0	0
3	2	2	1	2	1	1	1	1	1	1	0	1	1	0	0
2	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

Class 3

8	3	4	3	3	2	2	2	1	1	1	1	1	1	1	0
6	4	3	3	3	2	2	2	1	1	1	1	1	1	1	0
5	3	3	3	3	2	2	2	1	1	1	1	1	0	1	0
4	3	3	3	3	2	2	2	1	1	1	1	1	1	0	1
4	3	3	3	2	2	2	2	1	1	1	1	1	1	0	0
4	3	3	3	2	2	2	2	1	1	1	1	1	1	1	0
4	3	2	2	2	2	2	1	1	1	1	1	1	0	0	0
4	2	2	2	2	1	1	1	1	1	1	1	1	0	0	0
2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	0
3	2	2	2	2	2	1	1	1	1	1	0	1	0	0	0
3	2	1	1	1	1	1	1	1	1	0	0	0	0	0	0
2	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0
3	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
3	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0

Class 4

8	1	3	3	3	2	2	1	1	1	1	1	0	0	0	0
6	4	3	3	3	2	2	1	1	1	1	1	0	0	0	0
5	3	3	3	3	2	2	2	1	1	1	1	1	1	0	1
4	3	3	3	3	2	2	2	1	1	1	1	0	0	0	0
4	3	3	3	2	2	2	1	1	1	1	1	0	0	0	0
3	3	3	2	2	2	1	1	1	1	1	1	0	0	0	0
3	2	2	2	2	2	1	1	1	1	1	1	0	0	0	0
4	2	2	2	2	1	1	1	1	1	0	0	0	0	0	0
2	2	2	2	2	1	1	1	1	1	0	0	0	0	0	0
3	2	2	2	1	1	1	1	1	1	1	1	0	0	0	0
3	1	1	2	1	1	1	1	1	1	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
3	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
3	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0

Table II-3. Bit assignment tables for the 16x16 cosine transform coefficients for the four energy classes using the picture AIRPORT2 (Fig. III-3).

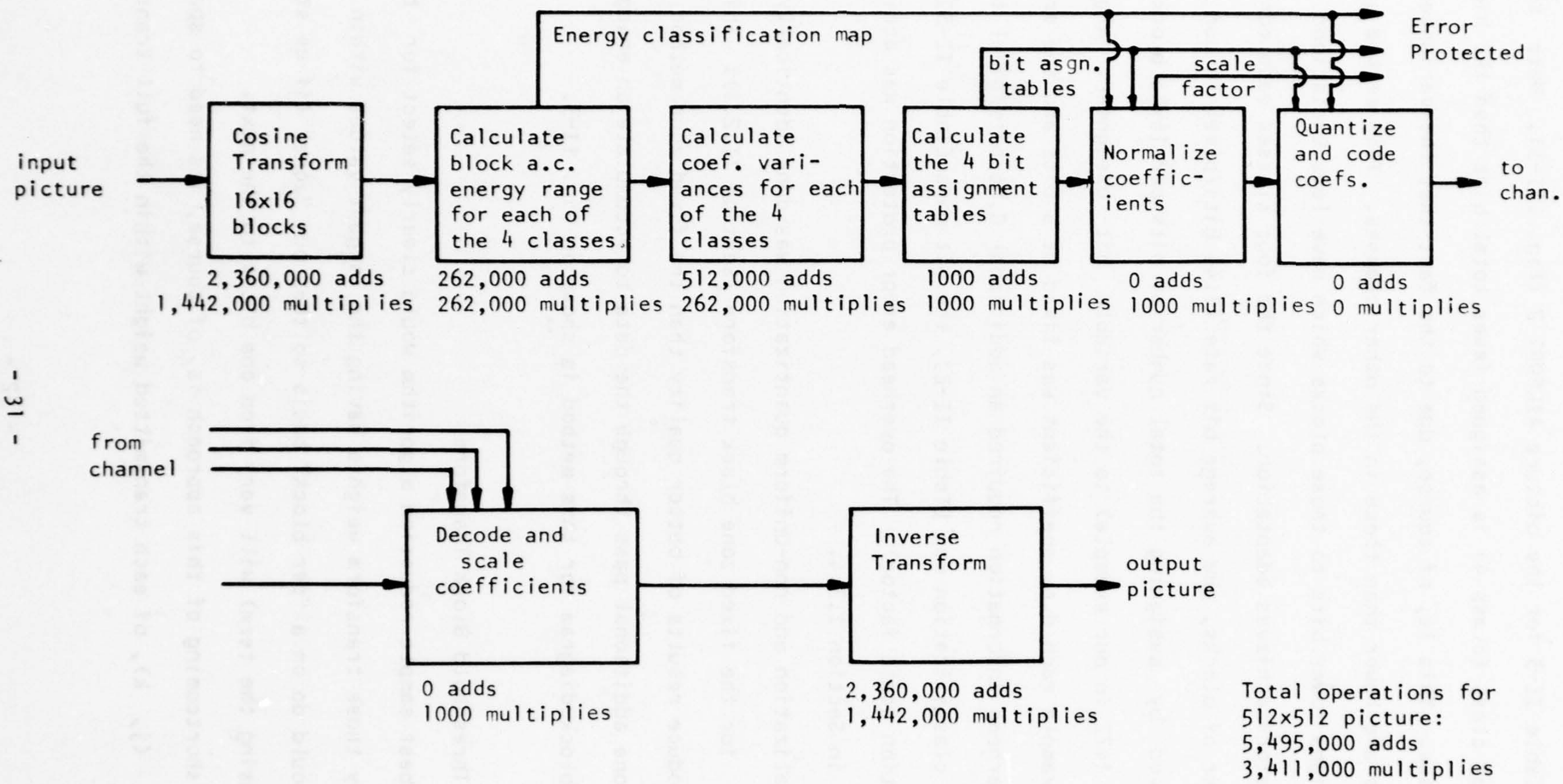


Fig. II-3: Four Class Zone Block Transform Coding
(Chen and Smith).

shown in Table II-3 for the picture AIRPORT 2 (Fig. III-3). Note that the low energy class (class 4) is assigned fewer total bits than the higher energy classes. This is, of course, due to the fact that the variances within this class are lower than those in the other classes. This method automatically assigns fewer bits to those blocks which have less variations in them and therefore achieves adaptation. Since the four classes each contain the same number of blocks, the average bit rate (1.44 bits/pixel in our example) is achieved by assigning the total number of bits per 16x16 block times 4 (this has 1472 in our example) to the variable bit assignment algorithms. In our example each d.c. coefficient was fixed at 8 bits and the error protected overhead information required an additional 0.06 bits/pixel (overhead uses the classification map [Table II-2], the bit maps [Table II-3] and the normalization scale factor). The overhead error protection was achieved as discussed in Section I.3.4.

Normalization and non-uniform quantization was done identically to that described for the fixed zone block transform (Section II.2.1). This method should produce results of better quality than the fixed zone method at the cost of one additional pass through the data to accumulate an energy histogram. A block diagram for this method is shown in Fig. II-3.

II.2.4. Threshhold Block Transform

The best sample reduction algorithm would clearly select for transmission only those transform weights having the largest values within a block. This it would do on a "per block" basis so that our "zone" (if we still insist on using the term) will vary from one block to the next.

The shortcoming of this approach is, of course, its need to specify the location, (j, k), of each transmitted weight within the full transform array. These locations are usually specified using some form of run length code, and at high compressions (below 1.5 bits/pixel) overhead becomes substantial.

The run length coding used in the threshhold algorithm tested here is based upon the coefficient ordering shown in Figure II-4. For the first ten coefficients within the ordering, run length code words were three bit quantities referring to the ordering distance between zero coefficients. Beyond this, seven bit code words specifying distances between nonzero coefficients were used. These code word lengths were chosen amongst several such lengths by trial and error (to achieve maximum compression) over several test pictures. Again the block size assumed is 16x16.

Since all transmitted coefficients are now guaranteed to be large, a fixed coefficient bit length of eight bits is assigned to all such transmissions. Once a threshhold sample reduction has been performed (based upon a desire to retain only the Y% largest transform weights*), a single variance of all chosen coefficients is calculated. This value is prepended to the picture transmission code and is used by the receiver to scale a single linear, eight-bit quantizer [15]. Again, a "times three" penalty is applied to these 8 bits of variance information to allow for error protection. See Figure II-5.

The run-length code words are interlaced with transform coefficient information during transmission. They place a heavy burden on the available channel bandwidth at the higher compression rates. (At 0.5 bits/pixel, run-length code words are responsible for almost 50% of the total information transmitted.) It is thus highly unlikely that this amount of control information can be heavily error protected. Hence, we have (in these early

*The choice of Y is, of course, dictated by the user-specified compression rate.

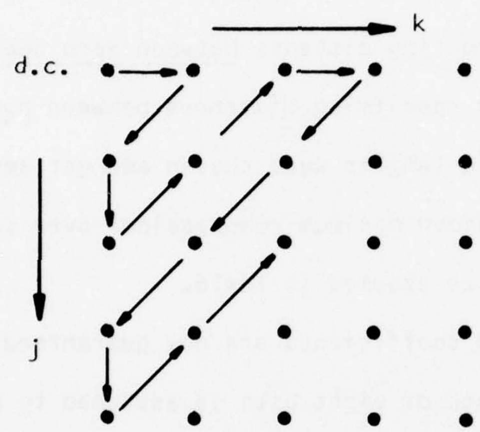


Fig. II-4: The transform coefficient ordering assumed for run length coding of threshhold sample reduced transform arrays. The first coefficient (d.c.) is always transmitted, regardless of its value.

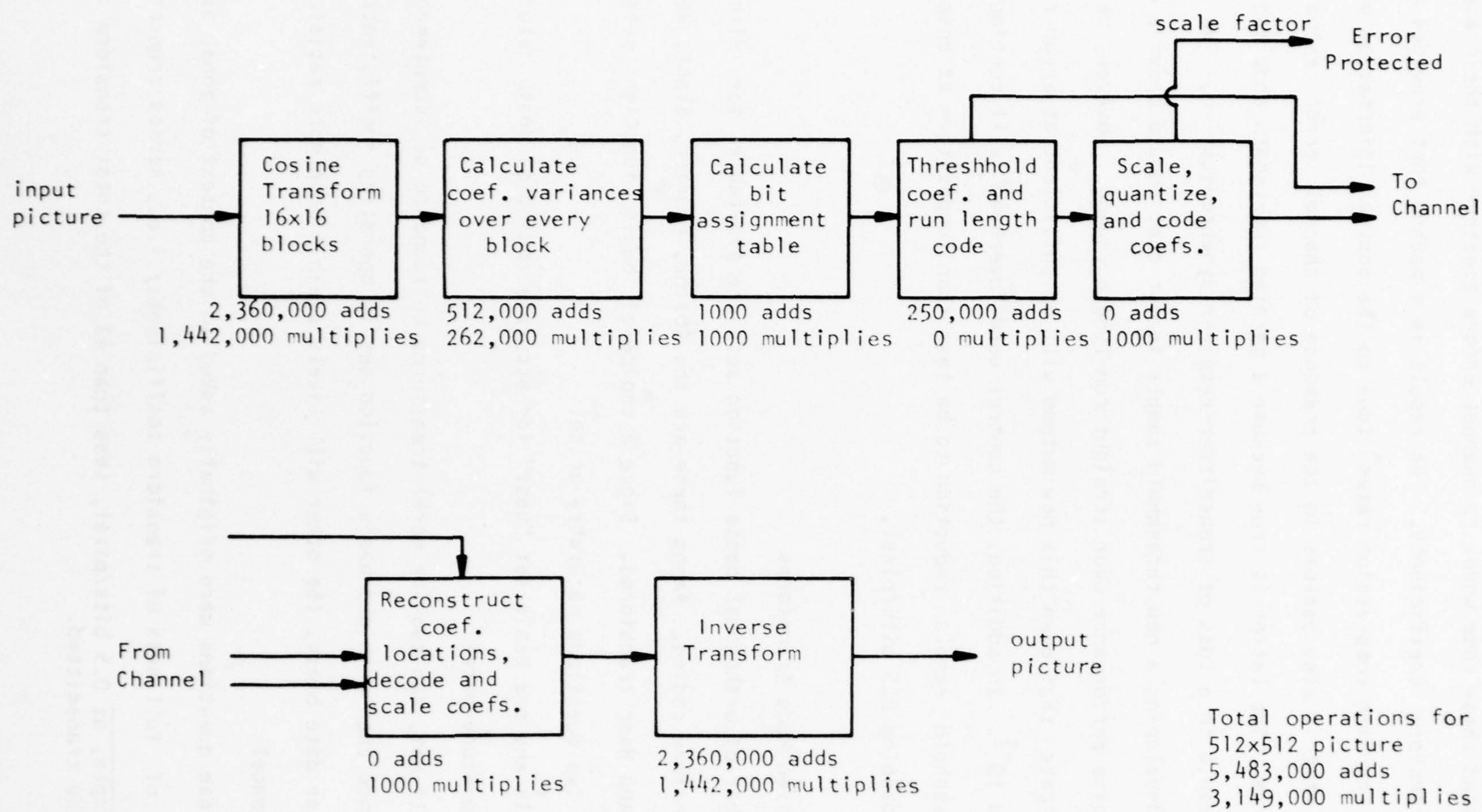


Fig. II-5: Threshhold Block Transform Coding tests) exposed these code words to channel errors together with their accompanying transform coefficients. The result is a method that provides poor quality at the high compression rates* (due to the control information overhead), and that is also useless in the presence of channel error rates of 10^{-3} or worse. The latter is true because a garbled run-length code can ultimately result in a loss of transmitter-receiver synchronization. We are currently developing a new threshhold sample format that should provide much improved noise performance over straight run-length coding. However, we do not anticipate that even this new method will be sufficient at error rates much beyond 10^{-3}. In addition, the control word "overhead" will continue to cause threshhold sample reduction to be less than competitive at compressions approaching 0.5 bits/pixel.

II.2.5. Mixed Mode Transforms

A number of orthogonal basis function sets are available for discrete space transform coding. Among these are the Cosine, Fourier, Slant, Walsh-Hadamard, and Haar transforms. Since a choice of basis function sets is available, two questions naturally arise:

(1) Is any one basis set "best" for accurate coding of most naturally occurring picture data?

(2) If not, can two (or more) transforms be found to be complementary in the sense that when one basis function set is operating inefficiently on a particular data block, the other will likely exhibit much more satisfactory performance?

Both of these questions were originally asked in the context of zonal sample reduction of full sets of transform coefficients; i.e., series truncation.

*For example, at 0.5 bits/pixel, less than 4% of the total transform array can be transmitted.

(Of course, one could make the same queries in a threshhold sample reduction context).

To answer these two questions, the above five transforms were applied to several pictures of size 256x256, blocked 16x16. Initial runs indicated that only the Cosine and Fourier transforms yielded any appreciable energy compaction into a small spectral zone based around "d.c." For example, the following table compares the number of times each of four transforms provided the most energy compaction (into a spectral zone of 31 out of 246 coefficients) over a picture having 256 16x16 data blocks. Such a sample reduction results in a data compression of 8.26:1. The picture in question was "AP1," an aerial photo of Midway airfield exhibiting a large amount of detail over most of the scene.

Transform Type	Number of times transform provided superior performance (out of 256 blocks)
Cosine	229 (89.5%)
Slant	12 (4.7%)
Walsh-Hadamard	9 (3.5%)
Haar	6 (2.3%)

To compare the cosine and Fourier transforms, the two methods were matched for energy compaction into a zone of 37 coefficients, for three separate photos all of size 256x256, blocked 16x16:

(1) AP1,

(2) AP4, another airport aerial photo exhibiting less detail than AP1, and

(3) GIRL, the standard "head and shoulders" view of a young girl.

The results are:

Transform Type	Picture AP1	AP4	GIRL
Cosine	213 (83.2%)	190 (74.2%)	245 (95.7%)
Fourier	43 (16.8%)	66 (25.8%)	11 (4.3%)

Thus, over a range of scenes, the cosine transform provides more effective energy compaction (than the Fourier) an average of 84.4% of the time.

Conclusions:

(1) The cosine transform is, by far, the most robust of the popular transform coding methods from an energy compression viewpoint.

(2) No other transform tested outperforms the cosine frequently enough to justify its use as an alternate "complementary" coding technique. Thus the mixed mode transform was not included in the subjective testing described in Section III.

II.3 Hybrid Coding

Habibi [7] has proposed a combination transform and predictive coder which has performed well in many applications. The version implemented performs a one-dimensional Cosine Transform across row sections. The transform coefficients are then coded using DPCM down the columns. We performed the discrete Cosine Transform over 32 long blocks in the horizontal direction (a length of 16 did not perform as well). Statistics collected on the transform coefficient difference (down the columns) were used to set the variable bit assignment (as described in Section II.1.1) used in the DPCM quantizer. The first row of the picture was transmitted using 6-bit PCM. In addition, to prevent too much propagation of channel errors due to the DPCM, every 64th row was also transmitted using PCM. The DPCM coding section is shown in Fig. II-6. The block diagram for the hybrid code is shown in Fig. II-7.

II.4. Space Domain Techniques

II.4.1. The CDC-MAPS Algorithm

This is a space-domain compression technique recently developed by Control Data Corporation for the Intelligence-Reconnaisance Division of RADC [10]. Initially, the approach examines a rectangular cluster of four pixels. If their various inter-pixel intensity differences are less than precribed threshholds, the four intensities are set equal to their mean value, thus merging this 2x2 group into a single sub-block for later transmission. If the local picture contrast is low, clusters of merged 2x2 sub-blocks can, in turn, be merged into a 4x4 sub-block. This process can be recursively applied (in low detail regions) up to any given maximum block size. Following [10], we have used a maximum size of 16x16.

The order of examination of pixels, for possible merger, is depicted in Figure II-8. As shown here, this same scanning pattern is followed when comparing previously merged sub-blocks for possible merger into even larger groupings.

The decision of whether or not to perform a merger of four pixels (or, later, four pixel groups) is based upon intensity differences among these pixels. As depicted in Figure II-9(a), the four pixel intensities are "middle step," "upper step," and "extreme" contrasts.

Each difference is compared with its corresponding threshhold in a "contrast control matrix." If all four differences do not exceed their respective threshholds, the merger attempt is successful: The four pixels

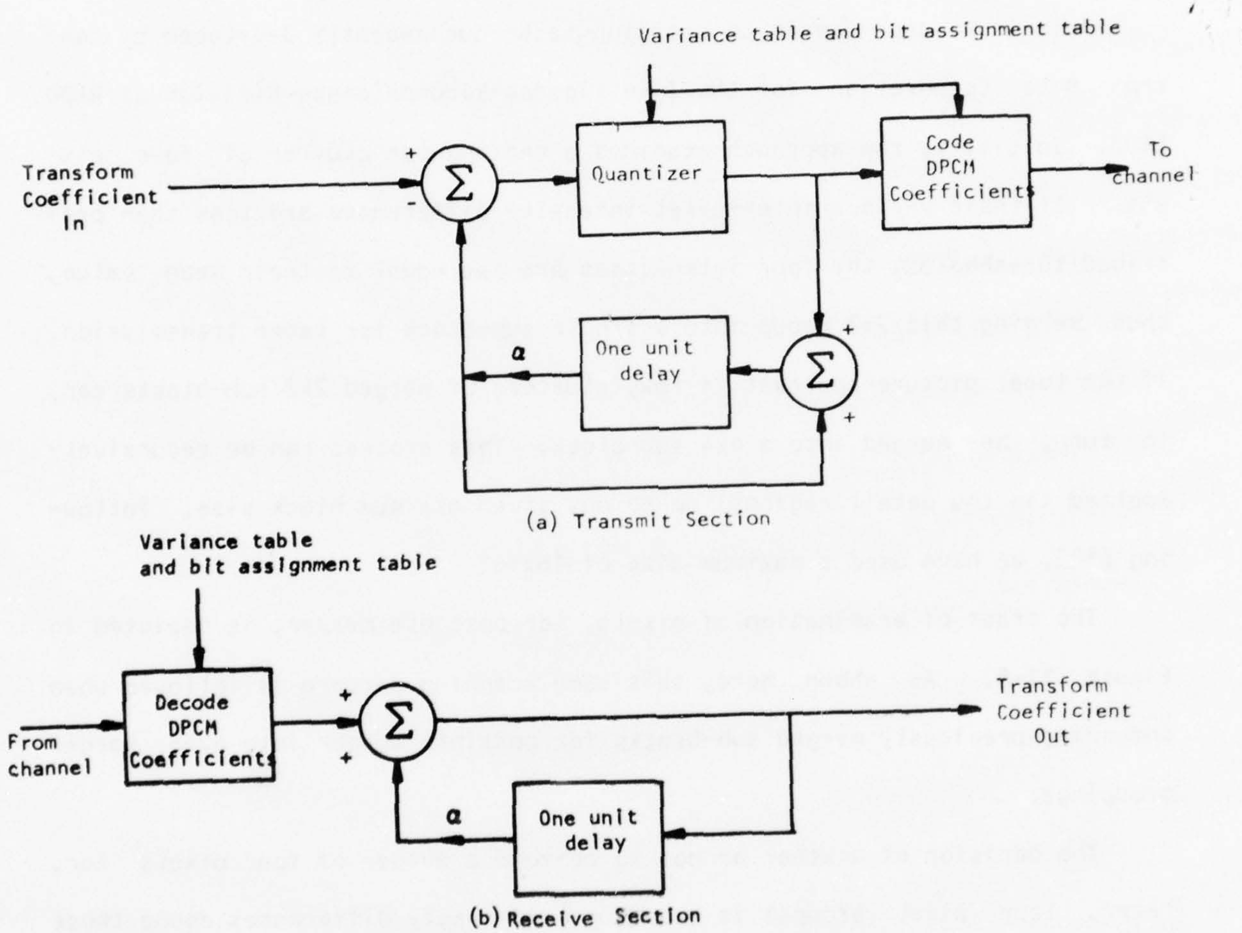


Fig. II-6: DPCM section of the hybrid coder. There is one such section for each of the 32 Cosine Transform coefficients.

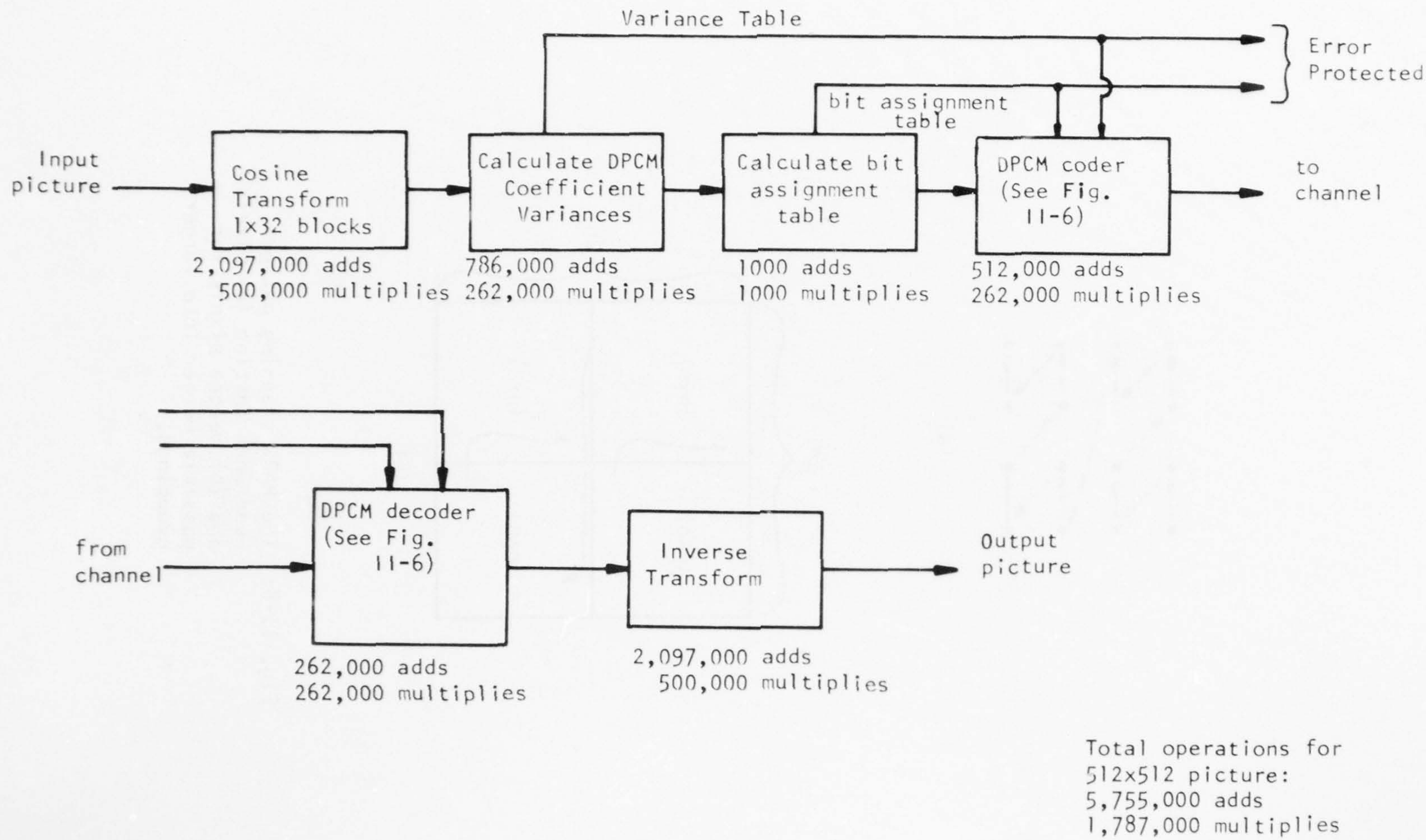
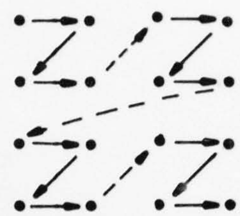
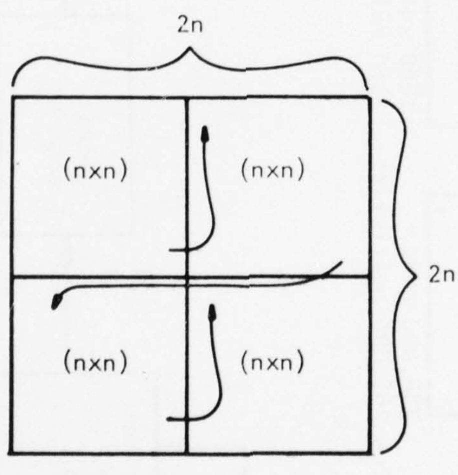


Fig. II-7: Hybrid Coding



(a)



(b)

Fig. II-8: The MAPS scanning pattern used when testing (a) pixels and (b) merged pixels for possible merger into larger groupings.

Let $I_3 > I_2 > I_1 > I_0$ be the intensities of four pixels (or groups of pixels) to be merged. Then:

I_0 — I_1: lower step

I_1 — I_2: middle step

I_2 — I_3: upper step

extreme $(I_3 - I_0)$

(a)

Merge attempt to size of:	Threshholds			
	Extreme	Middle	Lower	Upper
2 x 2	38	15	23	23
4 x 4	23	9	14	14
8 x 8	15	6	9	9
16 x 16	11	5	8	8

(b)

Fig. II-9: (a) the various step differences used in merge attempts.
(b) the contrast control matrix used in this work.

(or groups of pixels) are subsumed into a larger group which is characterized by the average value of its four components. Several threshholds appear in the contrast control matrix for each type of difference. Which threshhold is used depends upon the size of group that would result if the merge attempt were successful. Since merges into larger and larger groups will necessarily obscure more and more local detail, the difference threshholds used at the larger group size merge attempts become progressively smaller. That is, restrictions on mergers over wider areas of the picture are tighter. The contrast control matrix used during one of our tests* is given in Figure II-9(b). The matrices used by CDC to produce the results of [10] are not known at this writing. However, the values in Figure II-9(b) were obtained from an "initial guess" matrix by individually altering its rows and columns and observing the effects on the resulting bandwidth compression and picture quality. Following [10], within any row of the contrast control matrix, threshholds are chosen so that the "middle" step threshhold is smallest, and the lower and upper step threshholds are smaller than the extreme. The extreme, in turn, is usually smaller than the sum of the three separate step threshholds.

Of course, sooner or later a threshhold in the contrast control matrix will be violated and mergers within a particular region of a picture will have to stop. At this point picture data must be coded for transmission. In order to obtain an improvement over the CDC coding technique ([10], page 3-9), we have used the following coding format: A single three-bit code word precedes either a single word of intensity information or four words of intensity information. The code word indicates the size ($2^n x 2^n$) of grouping the succeeding information words will refer to and whether one or four in-

*This is picture SAM1 (Fig. III-1) at a compression of 1.5 bits/pixel.

tensities will follow. If only one intensity follows, a merger was apparently completed at size $2^n x 2^n$. On the other hand, four words will refer to intensities of four smaller groups of size $2^{n-1} x 2^{n-1}$. This method of coding is in contrast to the CDC practice of appending one two-bit control code to each and every intensity word.

The length of an intensity word used in this work was six bits. In addition, all three-bit control codes were lengthened to six bits in order to provide for a single error-correcting level of protection for this information. The factor of three penalty approach (Section I.3.4) was not used here due to the bulk of the control information needed to use the MAPS formating methods. Also, a unique, six-bit "start of block" control code was inserted into the transmitted bit stream in order to signal the start of a new 16x16 block. This has been considerable help in maintaining transmitter-receiver synchronization during transmission of MAPS variable length information frames over noisy channels. A simplistic block diagram is included as Fig. II-10.

II.4.2.1 Block Truncation Coding (BTC)

We now present a non-information preserving coding technique developed at Purdue which is very simple to implement, is well matched to the human observer, retains the resolution of the original, and results in a bit rate of between 1 and 2 bits/pixel [11].

One proceeds by first blocking the original picture into nxn blocks (we have used n=4 for our examples). Each block is then coded by producing a binary picture block where the two binary values are chosen such that the sample mean and variance are preserved [17].

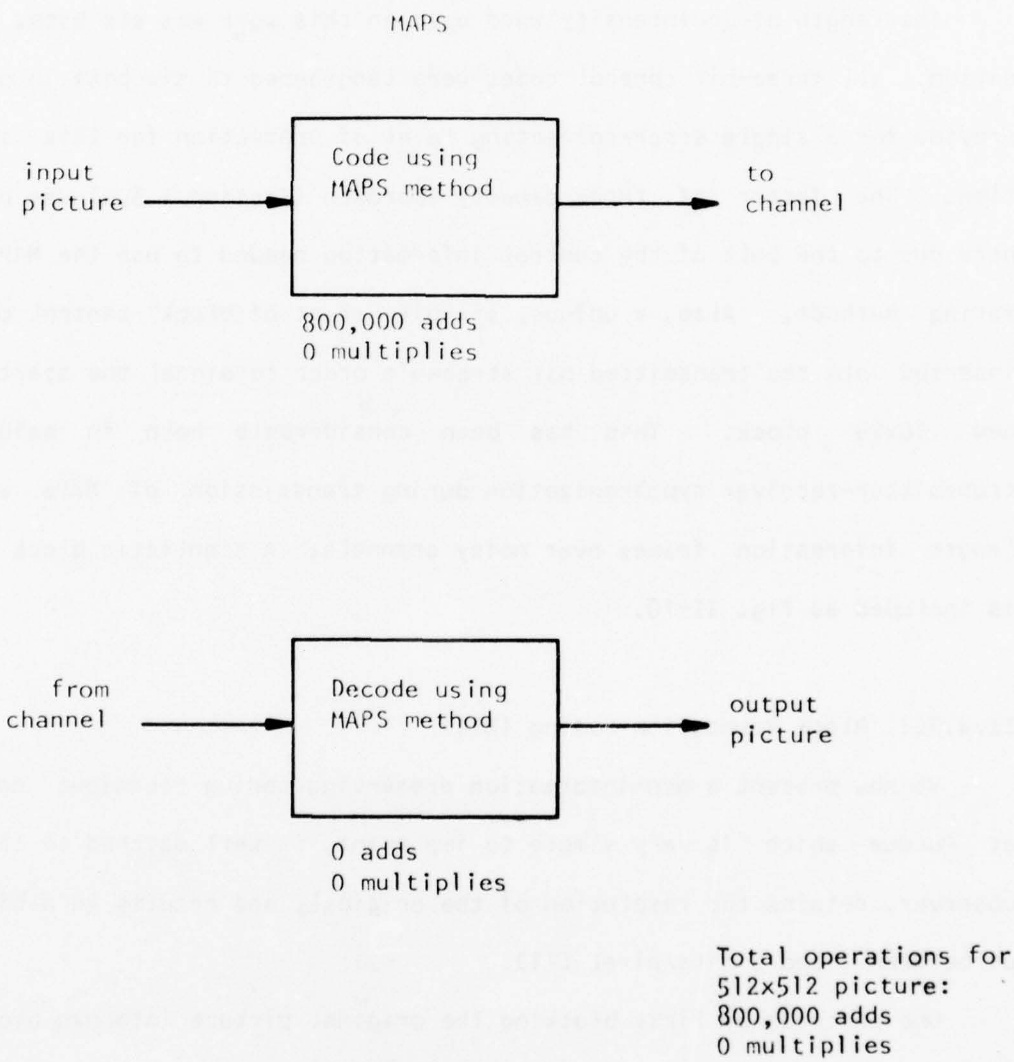


Fig. II-10: MAPS Coding arithmetic operations totals

Let $m=n^2$ and let $X_1, X_2, \ldots X_m$ be the values of the pixels in a block of the original picture.

Let

$$\overline{X}=\frac{1}{m}\sum_{i=1}^{m} X_i \text{ be the sample mean}$$

$$\overline{\sigma}^2 = \frac{1}{m-1}\sum_{i=1}^{m} (X_i-\overline{X})^2 \text{ be the sample variance} \qquad \text{(II-5)}$$

and let q = number of X_i's greater than $\overline{X}$.

The bit plane is chosen such that all pixel values above $\overline{X}$ are set to 1 and all other values are set to 0. The binary picture block is reconstructed from the bit plane and the mean and variance by setting all 0's to

$$a = \overline{X} - \overline{\sigma}\sqrt{\left(\frac{q}{m-q}\right)\left(\frac{m-1}{m}\right)}$$

and setting all 1's to

$$b = \overline{X} + \overline{\sigma}\sqrt{\left(\frac{m-q}{q}\right)\left(\frac{m-1}{m}\right)} \qquad \text{(II-6)}$$

Thus, the sample mean and sample variance in the binary picture block matches that in the original. The coded picture consists of a bit plane, $\overline{X}$, and $\overline{\sigma}$ for each nxn block of the picture. Assuming n=4 and 8 bits are used for $\overline{X}$ and $\overline{\sigma}$; this results in 2 bits/pixel. The receiver reconstructs the image block by calculating a and b and placing those values in accordance with the bits in the bit plane.

We have found that block boundaries are not visible in the reconstructed picture using this technique but two major artifacts do occur: (1) false contouring due to only two levels in each block, and (2) misrepresentation of some midrange points due to their assignment to either a high or low value. In the next section we will show methods for reducing these effects and reducing the bandwidth even further.

A few comments should be made about the appeal of this method. First is the obvious simplicity of the calculation involved. Only m pixels at a time need be considered, eliminating the need for picture storage and allowing real time coding with a small hardware device. Second is the suitability of this method to the human observer. The largest changes in a block are the ones coded. If no large changes are present, the most significant small variations are coded. The human is also insensitive to small variations in the presence of large variations, so that this technique is neglecting the very thing the human visual system is insensitive to. The third point is that the bit plane preserves the original accruacy of an edge or object location with no blurring. If anything, the effect is to enhance boundaries which is again suitable for human observation.

The problem of representing images by two levels has been addressed by other authors in the context of obtaining continuous tone displays using binary pictures [18-20]. The constrained average algorithm of Jarvis and Roberts [18] produces a binary rendition of an image; however, their method does not preserve the local mean and variance of the image. The constrained average method is used with two-level displays where any pixel is either "off" of "on." For BTC a pixel in the reconstructed image in a given subpicture is either a or b as given in Eq. (II-6), where a and b can take on any value from the original picture. The constrained average algorithm uses an adaptive threshholding technique at each pixel. This threshholdng technique is such that the threshhold is changed from pixel to pixel. The subpicture used is a moving window and not distinct blocks as with BTC. Lippel has presented results indicating the constrained average algorithm can be implemented by using a two-level dithering scheme [19], [21].

II.4.2.2 Example of BTC

First let us quickly review the BTC algorithm:

a) Image is divided in small non-overlapping blocks such as 4x4.

b) Sample mean $\overline{X}$, and sample variance $\overline{\sigma}^2$ are computed.

c) A bit plane is constructed such that each pixel location is coded as a "one" or a "zero" depending on whether that pixel is greater than $\overline{X}$.

d) The bit plane, $\overline{X}$, and $\overline{\sigma}$ are sent to the receiver.

e) The picture block is reconstructed such that $\overline{X}$ and $\overline{\sigma}$ are preserved. That is, pixels in the bit plane that are "0" are set to "a" and the "1"'s are set to "b" in equation II-6.
For example, suppose the following 4x4 picture block is given by:

$$X_{ij} = \begin{bmatrix} 121 & 114 & 56 & 47 \\ 37 & 200 & 247 & 255 \\ 16 & 0 & 12 & 169 \\ 43 & 5 & 7 & 251 \end{bmatrix}$$

so

$$\overline{X} = 98.75$$
$$\overline{\sigma} = 96.00$$
$$q = 7$$

and

$$a = 16.7 \approx 17$$
$$b = 204.2 \approx 204$$

the bit plane is:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The reconstructed block becomes:

$$\begin{bmatrix} 204 & 204 & 17 & 17 \\ 17 & 204 & 204 & 204 \\ 17 & 17 & 17 & 204 \\ 17 & 17 & 17 & 204 \end{bmatrix}$$

and the sample mean and variance are preserved. A block diagram of the coding method is shown in Fig. II-11.

II.4.2.3 Some Other Considerations of BTC

Block Truncation Coding can be formulated as a problem of obtaining an adaptive two-level (one-bit) quantizer such that the visible properties of the image are preserved. If one could obtain a fidelity criterion for an image that would somehow represent the desired properties of an image (i.e., edge preservation, no false contouring, texture preservation), then one could use rate-distortion theory to obtain a one-bit quantizer to match this fidelity criterion [14], [22]-[27]. This is indeed a herculean effort and was not pursued in this work.

For the work presented here, the fidelity criterion used in the quantizer in one of preserving the mean and variance. This criteria seems to work well in most situations when images must be either transmitted or stored. Other criteria have been used in testing BTC--in particular, minimizing the mean square error [15] or the mean absolute error [28]. To use these criteria, it is necessary to fit the one-bit quantizer to the data by an exhaustive search. One can obtain poorer results by making an ad hoc de-

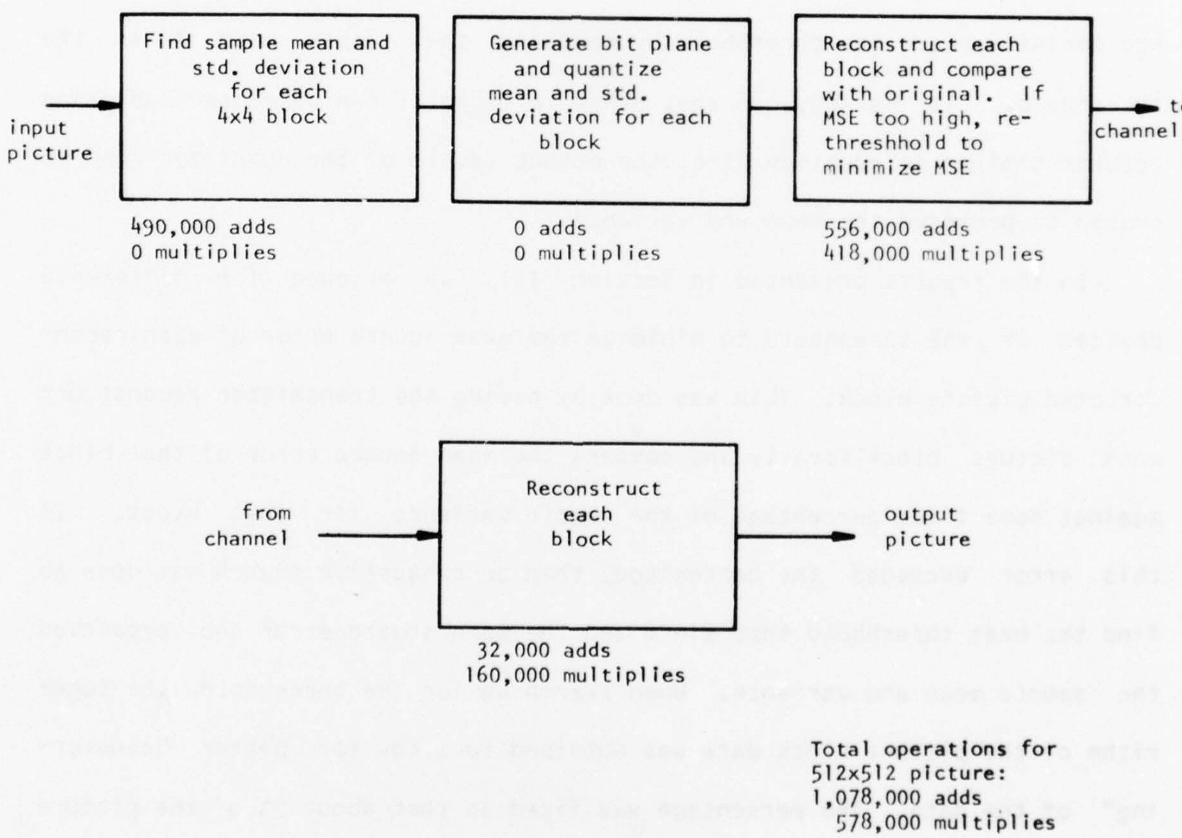


Fig. II-11: Block Truncation Coding

cision as to what the probability density function of the image is. It is possible to obtain quantizers that are robust for large classes of density functions, but these techniques have not been widely applied to imagery data [29].

The advantages of a fidelity criterion that only preserves the mean and variance is that it is not necessary to do an exhaustive search to match the fidelity criterion. In general if we are interested in fitting a one-bit quantizer to data in each picture block, it is necessary to obtain three parameters: the two output levels and the quantizer threshhold. In using the basic BTC algorithm as discussed section II.4.2.2, we are making an ad hoc decision about the threshhold by choosing the sample mean $\overline{X}$ as the threshhold. It is obvious that other threshholds can be chosen and using results similar to equation II-6, the output levels of the quantizer can be chosen to preserve the mean and variance.

In the results presented in Section III, we allowed for difference choices of the threshhold to minimize the mean square error of each reconstructed picture block. This was done by having the transmitter reconstruct each picture block locally and compare the mean square error of that block against some fixed percentage of the sample variance for that block. If this error exceeded the percentage, then an exhaustive search was done to find the best threshhold that minimized the mean square error and preserved the sample mean and variance. When searching for the threshhold, the logarithm of the picture block data was obtained to allow for better "clustering" of the data. The percentage was fixed sc that about 5% of the picture blocks in each scene were searched exhaustively for a new threshhold. Although this did increase the computational load at the transmitter, the receiver reconstruction procedure <u>was not</u> affected.

Experimental evidence has indicated that it is necessary to code the mean and variance (actually the standard deviation) with less than 10 bits. This allows for considerable savings and no perceivable errors upon reconstruction. This then gave a compression rate of 1.5 bits/pixel. To allow for more savings in coding, we looked at various ways of coding the bit plane. The entropy of the bit plane has been approximated by calculating th entropy of 8-bit sections of the bit planes. Typical values indicate that it is necessary to allow 0.85 bits/pixel in the bit plane instead of the nominal 1 bit/pixel. The coding of the bit plane could be implemented by using standard minimum redundancy coding such as Huffman coding [30]-[32]. Due to the poor performance of these codes in the presence of noise and the fact that gain in compression would not be that great, (not to mention the overhead in coding), the bit plane was not entropy coded for the results presented in Section III. We do intend to pursue robust coding methods for the bit plane.

In developing BTC to this point in time, we have looked at various other techniques to improve the performance such as dithering [33] and "soft" threshholds. These methods were not used for the results presented in Section III because of the extra computational overhead and/or the results did not greatly improve the reconstructed picture [11].

II.4.3 Image Modelling and DPCM

A 2-dimensional image model is formulated in this section using a seasonal autoregressive time series. With appropriate use of initial conditions, the method of least squares is used to obtain estimates of the model parameters. The model is then used to regenerate the image. A differential pulse code modulation (DPCM) scheme is also investigated for future use in comparing DPCM with other coding methods.

II.4.3.1 Introduction

In recent years much work has gone into obtaining reasonable spatial models for images [34, 35, 36, 52]. This has led to some very good results in areas such as image enhancement and coding [37-41]. The problem still exists, however, of obtaining a good model for an image. In this context various studies of the nature of the human observer in imagery systems have been undertaken [14, 27].

In this section image modelling is approached from the point of view of classical time series analysis using the method of least squares [42, 43]. A seasonal one-dimensional model [42] is obtained and used to regenerate test images. This approach is shown to have some promise in the area of texture or background modelling. Similar results have been presented by McCormick and Jayaramanurthy using a seasonal autoregressive integrated moving average model [51]. The model is also demonstrated in a differential pulse code modulation scheme (DPCM).

II.4.3.2 The Image Model

Suppose each NxN discrete image is described as a matrix $y(i,j)$. If the image is assumed to be a sample picture from a two-dimensional discrete homogeneous Gaussian-Markov field, one can show that the pixel (picture element) at $y(i,j)$ can be represented as described in [44, 45]:

$$y(i,j) = \sum_{m,n\neq 0}\sum \theta_{mn}\, y(i-m,j-n) + u(i,j) \qquad \text{(II-7)}$$

for all (m,n) such that $(i-m,j-n)\epsilon D$ and where

1) $D = \{(i,j) ; i < a \text{ or } j < b\}$

2) $E[u(i,j)\ y(i-m,j-n)] = 0 ; m,n \geq 0, m+n > 0$

3) $E[u(i,j)u(k,\ell)] = \sigma^2 \delta_{ik}\delta_{\ell j}$

4) $E[u(k,\ell)] = 0$

It is easily recognizable that the region D consists of the first a rows and b columns of the picture. This region consists of the initial conditions for the image. There are more general regions for D that include more neighbors but the problem of obtaining accurate parameter estimates is more complicated [50]. The noise u(i,j) is assumed to be a zero mean independent, identically distributed Gaussian process. This model says that the picture can be generated by driving a two-dimensional recursive digital filter with Gaussian white noise.

The special case of a Gaussian-Markov field with a and b equal to one is sometimes assumed for the picture:

$$y(i,j) = \theta_1 y(i-1,j) + \theta_2 y(i-1,j-1) + \theta_3 y(i,j-1) + u(i,j) \tag{II-8}$$

The region D then consists of the first row and column of the picture. A model could also be formulated so that y(i,j) will depend on y(i-1,j+1). This model will be presented in Section II.4.3.3. By row-concatenation one can obtain a one-dimensional formulation of equation (II-8):

$$\text{let} \quad (i,j) \rightarrow k = (i-1)N+j$$

$$y(k) = \theta_1 y(k-N) + \theta_2 y(k-N-1) + \theta_3 y(k-1) + u(k) \tag{II-9}$$

This formulation of y(•) is suggestive of horizontal line scanning. This transformation could also be used to obtain a vertical scan type of formulation. This formulation is shown in Section II.4.3.3 to work quite well for parameter estimation of the θ's and σ^2. A slightly different approach can be used to formulate a true scanned image using cyclostationary random processes [35, 46].

Except for the handling of the initial conditions, Eq. (II-9) can easily be recognized as a seasonal autoregressive time series [42]. One can obtain a more compact version of equation (II-9):

$$y(k) = \underline{\theta}^T \underline{Z}(k-1) + u(k) \tag{II-10}$$

where

$$\underline{\theta}^T = [\theta_1, \theta_2, \theta_3]$$

$$\underline{Z}^T(k-1) = [y(k-N), y(k-N-1), y(k-1)]$$

The vector $\underline{Z}(k-1)$ is called the "past history" of the process [10].

Throughout the rest of this section, the image will be assumed to be described by the Gaussian-Markov field of equation (II-7). Through the change of variables discussed above a seasonal autoregressive time series will be obtained for various models described by equation (II-7).

II.4.3.3. Parameter Estimation

To fit the model given in equation (II-10) to a particular picture one has to obtain estimates of the regression parameters $\underline{\theta}$ and white noise variance σ^2. The method of least squares was chosen to obtain the parameter estimates for the model. Given equation (II-10) one wishes to find $\underline{\theta}$ and σ^2 such that the squared error between the actual pixel value and the pixel value generated by the model is minimum. The squared error is given by

$$J_N(\underline{\theta}) = \sum_{\substack{k=1 \\ k \neq I.C.}}^{N^2} (y(k) - \underline{\theta}^T \underline{Z}(k-1))^2$$

where I.C. = {y(k) derived from pixels $(i,j) \in D$} (II-11)

It is easily shown that the values of $\underline{\theta}$ and σ^2 which minimize $J_N(\theta)$ are

$$\hat{\underline{\theta}}(N_1) = \left[\sum_{\substack{k=1 \\ k \neq I.C.}}^{N^2} (\underline{Z}(k-1)\underline{Z}^T(k-1)) \right]^{-1} \sum_{\substack{k=1 \\ k \neq I.C.}}^{N^2} y(k)\underline{Z}(k-1)$$

$$\hat{\sigma}^2 = \frac{1}{N_1} \sum_{\substack{k=1 \\ k \neq I.C.}}^{N^2} (y(k) - (\hat{\underline{\theta}}(N^2))^T \underline{Z}(k-1))^2$$

where $N_1 = N^2 - I.C.$ (II-12)

Since it is assumed the process $u(\cdot)$ is Gaussian, the results obtained in equation (II-12) are the same as the conditional maximum likelihood estimates of $\underline{\theta}$ and σ^2 [42].

The term

$$w(k) = y(k) - (\hat{\underline{\theta}}(N_1))^T \underline{Z}(k-1) \quad \text{(II-13)}$$

is called the residual of the model. If the image is actually described by equation (II-10) and if $\underline{\theta} = \hat{\underline{\theta}}(N_1)$, then w(k) would of course be the white noise driving process. In practice a complex image cannot fully be described by a simple model as in equation (II-10) and therefore the residuals can be used as an indication of how good the model actually fits the particular picture. In general then, it would be desirable that the residuals represent a zero-mean white Gaussian sequence. Various tests can be performed on the residual sequence to verify the above properties [42, 43, 51]. One very simple test that can illustrate the validity of the model is to let $\underline{\theta} = \underline{\hat{\theta}}(N_1)$ and $\sigma^2 = \hat{\sigma}^2$. Then regenerate the image according to equation (II-10) with the initial conditions and an independent Gaussian random number generator. The quality of the regenerated image of course would be the final test.

II.4.3.4 Some Experimental Results

The following two Gaussian-Markov models were used for this study

$$y(i,j) = \theta_1 y(i-1,j) + \theta_2 y(i-1,j-1) + \theta_3 y(i,j-1) + u(i,j) \tag{II-14}$$

$$y(i,j) = \theta_1 y(i-1,j) + \theta_2 y(i-1,j-1) + \theta_3 y(i,j-1) + \theta_4 y(i-1,j+1) + u(i,j) \tag{II-15}$$

The model of equation (II-14) was discussed in Sections II.4.3.2 and II.4.3.3. The model of equation (II-15) is slightly different but can also be formulated into a seasonal autoregressive time series similar to equation (II-10). The initial conditions for the latter model are the first row and the first and last column of the image.

Due to the gross non-homogeneous nature of large images it was decided to separate the image into NxN subpictures and to fit the same model type to each subpicture of a large picture. In other words, all the models for each large image had the same form but a different parameter set $\underline{\hat{\theta}}(N_1)$ and $\hat{\sigma}^2$ for each subpicture.

Results were obtained for two large pictures each of which were 256x256 pixels. The subpicture size was chosen to be 16x16 pixels, so that each 256x256 scene had 256 parameter sets associated with it. No attempt was made to mix different model formulations within a given scene. After the parameter set was obtained each image was regenerated as described in Section II.4.3.3 with a Gaussian random number generator (provision was made for the cases where the residual was not zero-mean). Results are shown in Figures II-12 and II-13 for both models described by equations (II-14) and (II-15). The absolute difference picture between the original and regenerated image was also obtained as well as the root mean square (rms) error. The texture image is that of cork.

The results obtained are interesting in that the picture was generated from the initial conditions and the parameter set along with the model description and a random number generator. The results obtained for the model described by equation (II-15) are much better than that described by equation (II-14). One should remember that in each subpicture the initial conditions used were the actual pixel values for those rows and columns respectively.

There is no guarantee using the method of least squares that the model obtained will be stable. In fact, in the girl's face scene for the model described by equation (II-14) one can see subpicture blocks which have an unstable model formulation. One can observe this by noticing the white and black streaking in the regenerated scene toward the bottom of the picture.

The two scenes presented in Figures II-12 and II-13 consist predominantly of low spatial frequencies. The models described by equations (II-14) and (II-15) are of course low-pass in nature.

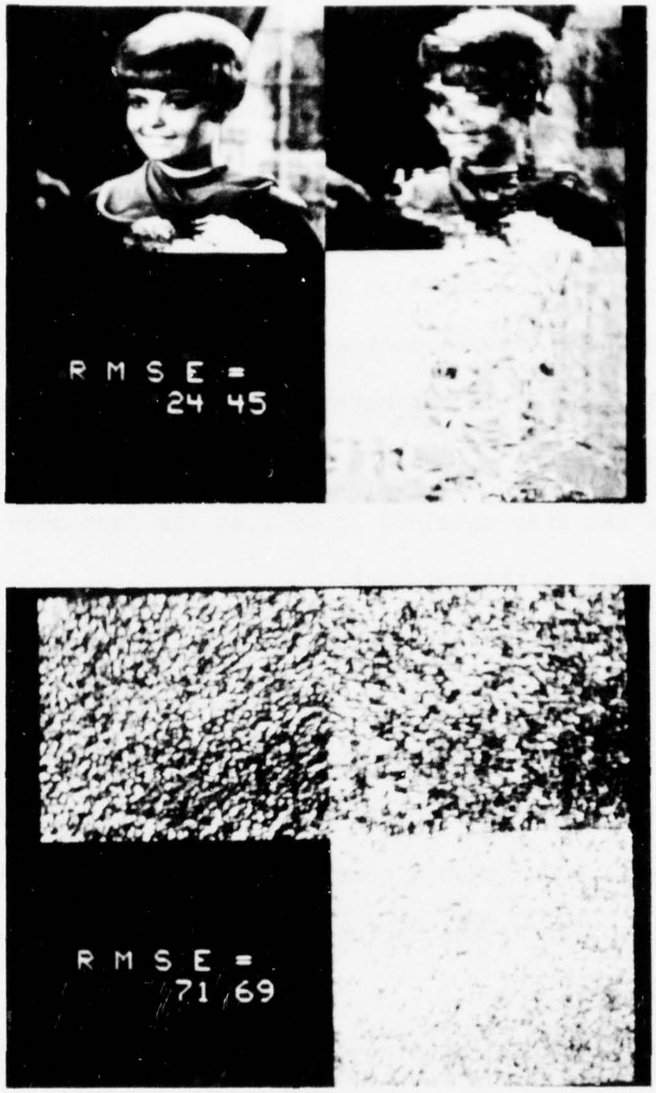


Figure II-12. Upper Left: Original Image; Upper Right: Regenerated Image; Lower Right: Absolute Difference Picture (Zero error corresponds to medium gray). Image regenerated using the model described in (8). Lower left is the rms error.

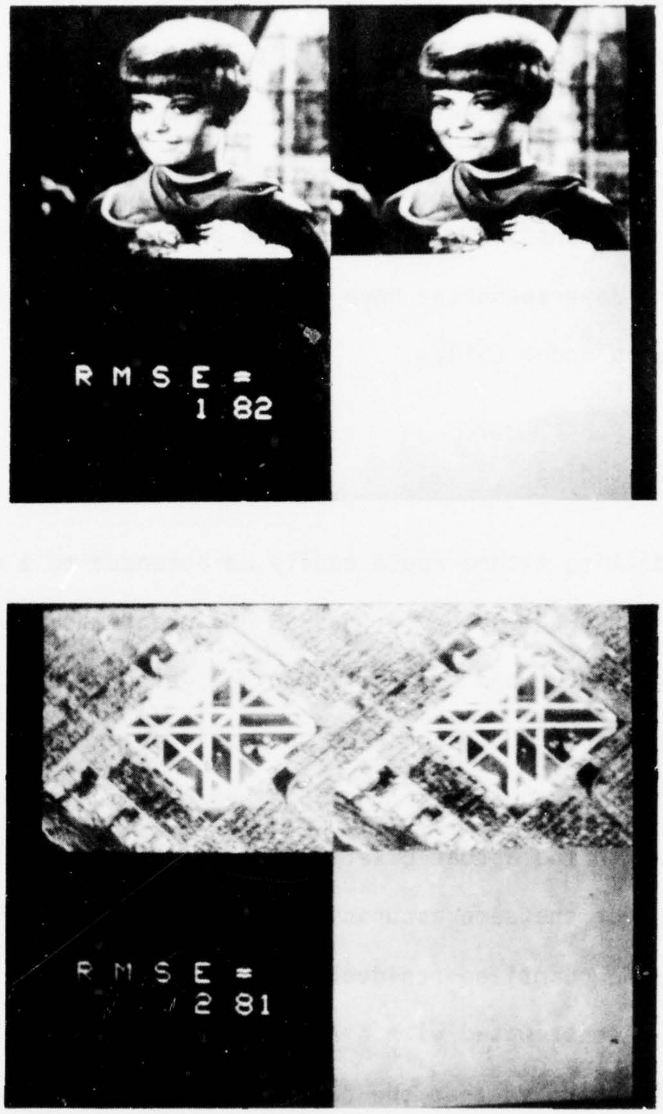


Figure II-13. Upper Left: Original Image; Upper Right: Regenerated Image; Lower Right: Absolute Difference Picture. Image regenerated using the model described in (9). Lower left is the rms error.

These results suggest that Gaussian-Markov models could work quite well with textures and background scenes. In an image coding scheme the texture model would only have to be transmitted and the receiver could regenerate the texture locally. In fact one would not have to transmit the entire set of initial conditions, this could lend to compressions of 10:1 or higher. McCormick and Jayaramanurthy have obtained similar results using a slightly more complicated model [51].

II.4.3.5 DPCM Coding

This modelling method could easily be extended to a differential pulse code modulation (DPCM) scheme. In DPCM one quantizes the difference between an actual pixel value and a predicted pixel value [41, 47, 49, 51]. The above difference could of course be the residual equation (II-13). The philosophy of this arrangement is that one needs to assign fewer bits to the residual than to the actual pixel value, due to the inherit low-variance of the residual, for the same accuracy to achieve the same quality reconstruction. Given the quantized residual and the model coefficients, the original image can be reconstructed with a reasonable degree of accuracy.

It should be noted that the residual cannot be evaluated and quantized directly, this would lead to an "open-loop" DPCM scheme and the quantization errors would propagate. A feedback quantizer is generally used in DPCM to keep the quantization errors from propagating [47]. The feedback quantizer is shown in Figure II-14.

One has to make some sort of choice as to what kind of quantizer should be used. Recently some work has been done in the area of optimal quantizers in the context of a human visual system model [48]. These quantizers are designed so that quantization noise artifacts such as slope overload, false

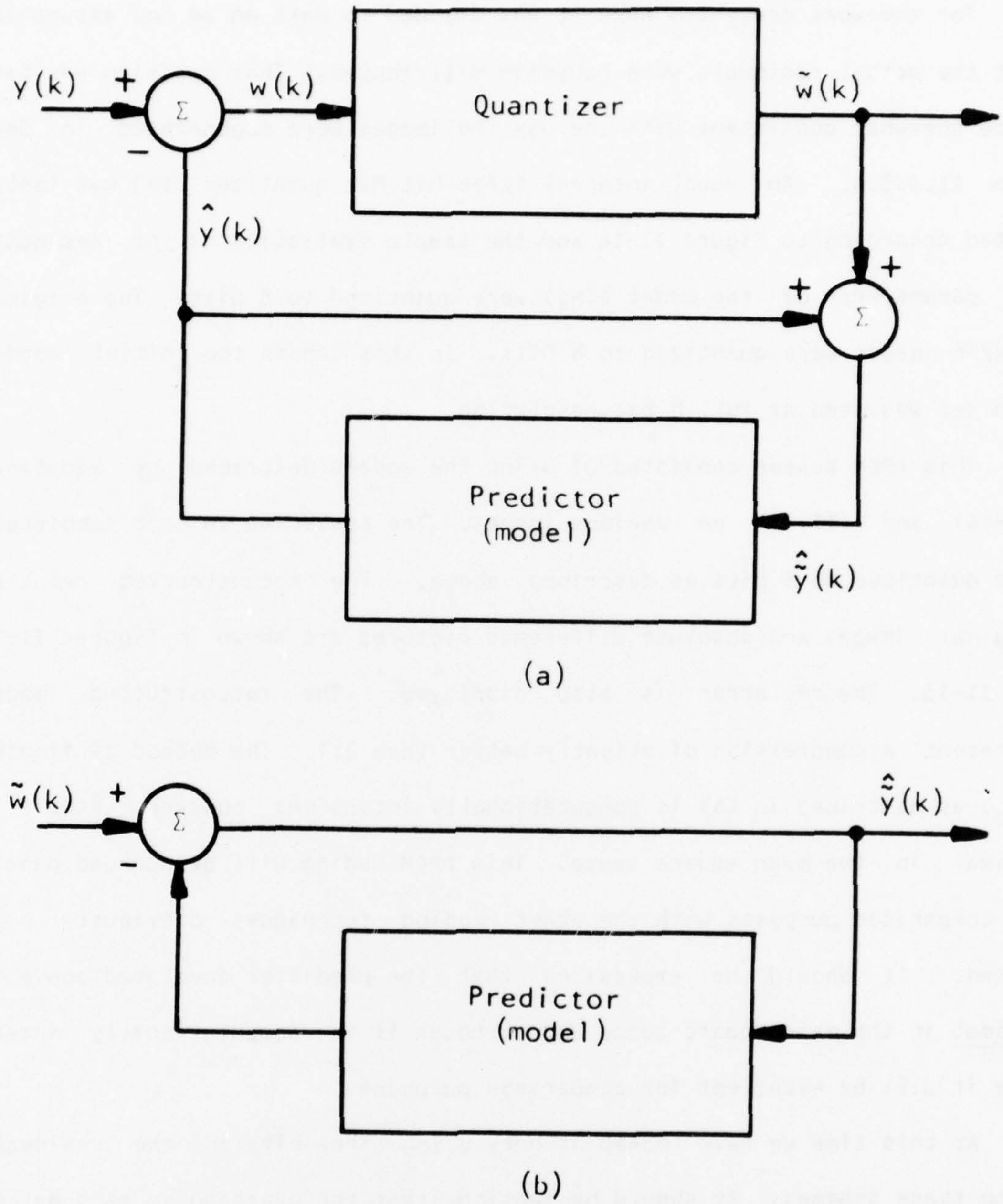


Figure II-14. (a) Feedback DPCM Quantizer; (b) Regeneration Scheme. $\tilde{x}$ indicates $x(\cdot)$ quantized and $\hat{x}(\cdot)$ indicates $x(\cdot)$ predicted.

contouring, and edge busyness are minimized.

For the work presented here it was decided to make an ad hoc assumption that the actual residuals were Gaussian distributed. This decision was made to be somewhat consistent with the way the images were regenerated in Section II.4.3.4. An equal interval three bit Max quantizer [15] was implemented according to Figure II-14 and the sample statistics of the residual. The parameters of the model $\underline{\hat{\theta}}(N_1)$ were quantized to 8 bits. The original 256x256 images were quantized to 8 bits. In this scheme the initial condition set was used at full 8 bit resolution.

This DPCM scheme consisted of using the models described by equations (II-14) and (II-15) on various images. The residuals in each subpicture were quantized to 3 bits as described above. The reconstructed results, original images and absolute difference pictures are shown in Figures II-15 and II-16. The rms error is also displayed. The reconstructed images represent a compression of slightly better than 2:1. The method of finding $\underline{\hat{\theta}}(N_1)$ as described in (6) is computationally intensive; <u>however</u>, $\underline{\hat{\theta}}(N_1)$ is optimal in the mean square sense. This DPCM coding will be pursued mainly for comparison purposes with the other coding techniques previously mentioned. It should be emphasized that the predictor developed above is <u>optimal</u> in the mean square sense and although it is computationally intensive it will be excellent for comparison purposes.

At this time we have looked at only using three bits for the residuals using these schemes. It should be mentiond that the overhead in this method with the initial conditions is quite high and we are planning on reducing the initial condition set to allow higher compressions. We will also examine the effects of using less bits for the residuals and look at channel errors with this system to compare with our previous results and methods dis-

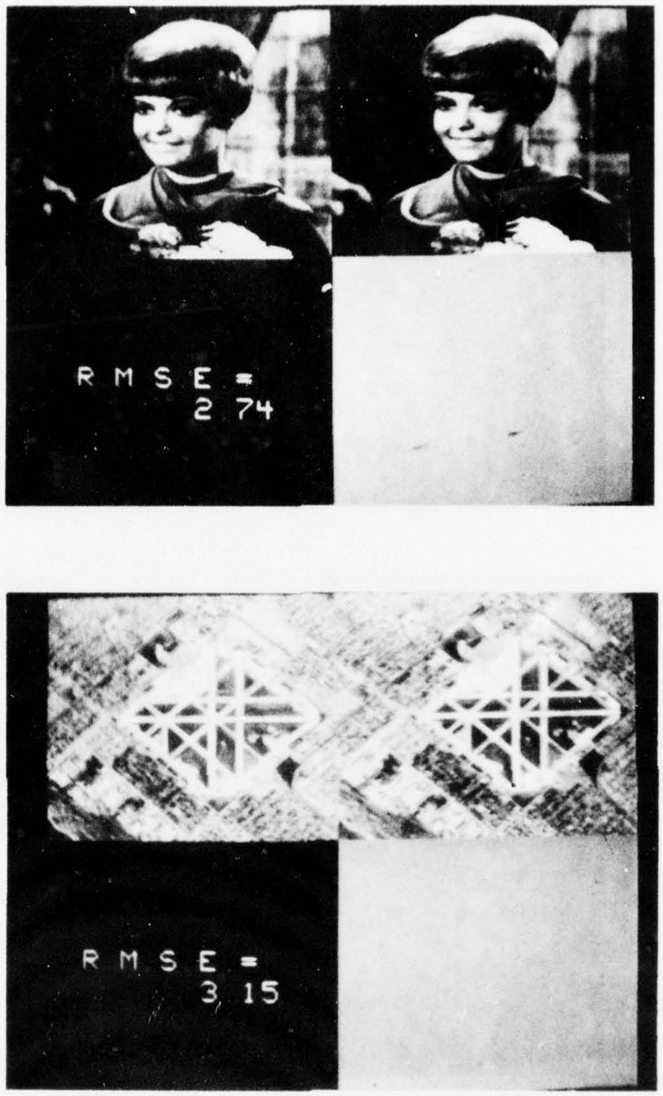


Figure II-15. Upper Left: Original Image; Upper Right: Reconstructed Image; Lower Right: Absolute Difference Picture. Image reconstructed using the model described in (8) and DPCM scheme. Lower left is the rms error.

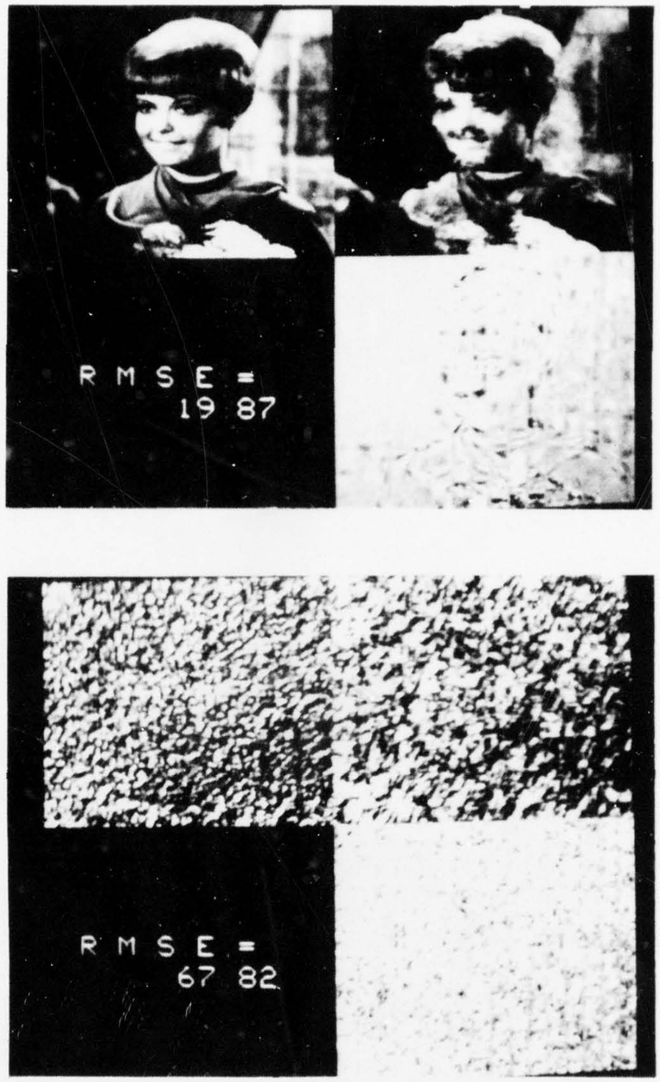


Figure II-16. Upper Left: Original Image; Upper Right: Reconstructed Image; Lower Right: Absolute Difference Picture. Image reconstructed using the model described in (9) and DPCM scheme. Lower left is the rms error.

cussed above. Experimental evidence indicates that a Laplacian and not a Gaussian density function should be used for the residuals [53].

III. Results

III.1. Original Images

Three 512x512 images were selected as originals. SAM1 (Fig. III-1) and SAM2 (Fig. III-2) were supplied by RADC and are pictures of part of the Northeast Test Site Area. SAM1 is a missile site (SAM) with both fully equipped and empty launching sites. Missile crates and mobile radar equipment are also visible. SAM2 includes an AAA missile site, a forested region, and several tanks. Both pictures are quantized to 6 bits (64 grey levels).

The third original, AIRPORT2(AP2), Fig. III-3, has been quantized to 8 bits (256 levels). This airport scene was chosen due to its wide variability of image characteristics. Of interest are the city areas, the airport runways, and boats and other small objects in the water. AP2 is a much wider bandwidth image than the other two (adjacent pixels frequently exhibit very little correlation).

To help evaluate coding errors, the raw PCM original data was subjected to the same random error rates as the coded pictures. A sample for AP2 at 10^{-3} probability of error is shown in Fig. III-4.

III.2. Reconstructed Results

Examples of reconstructed pictures using all the tested coding methods follow. However, we first show three "difference pictures" between the original and the results of three coding methods. These are Figs. III-5 to III-7. The coding techniques used were block truncation, Chen and Smith, and MAPS, respectively. A negative difference is displayed as darker and a positive difference as lighter than gray which represents no difference. Note the stronger correlation of the coding error with the original image in

Fig. III-1: Original photo "SAMI", 512 x 512 pixels, 6 bit quantization, photo number 109.

Fig. III-2: Original photo "SAM2", 512 x 512 pixels, 6 bit quantization, photo number 119.

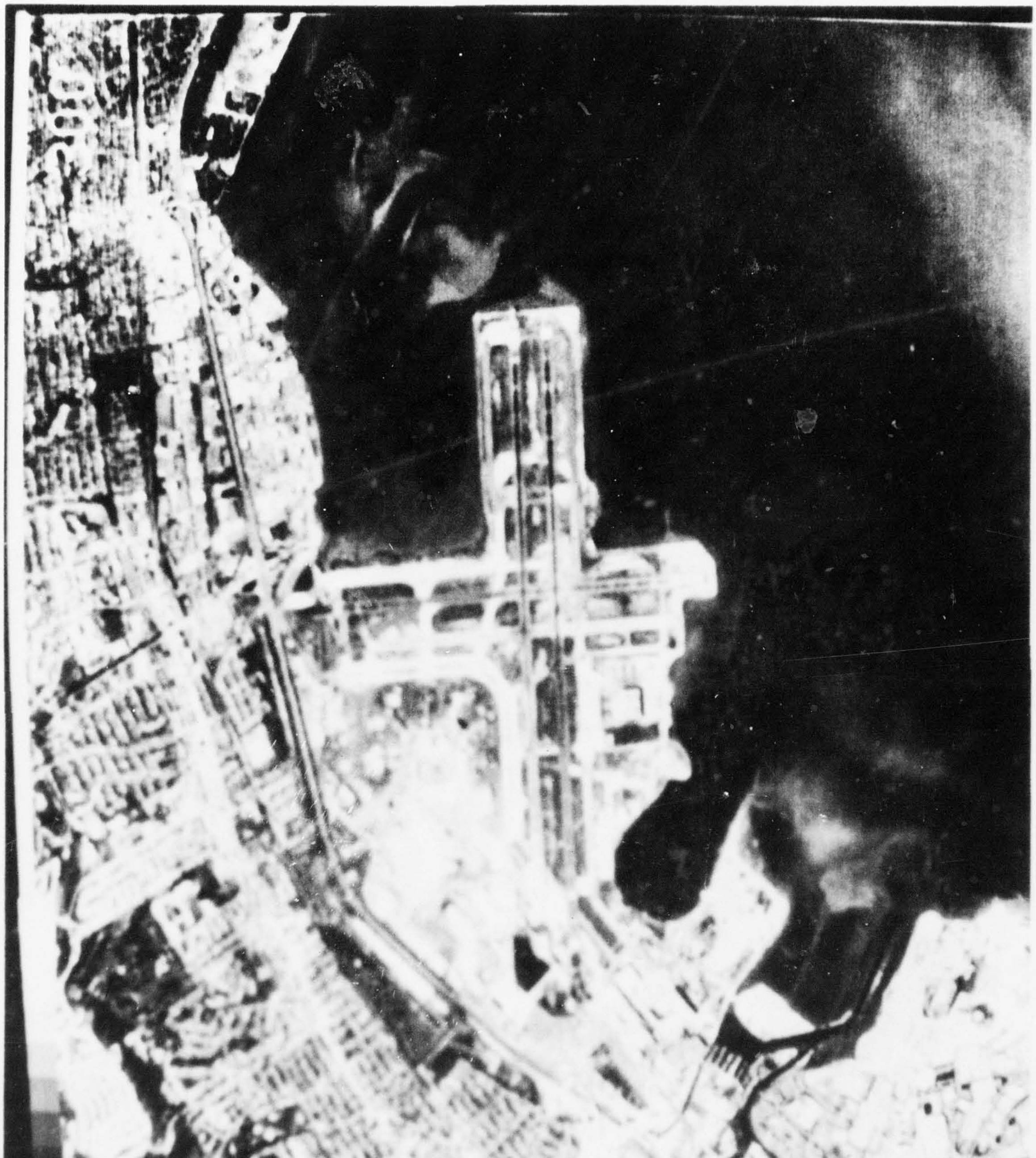


Fig. III-3: Original photo "AP2", 512 x 512 pixels, 8 bit quantization, photo number 300.

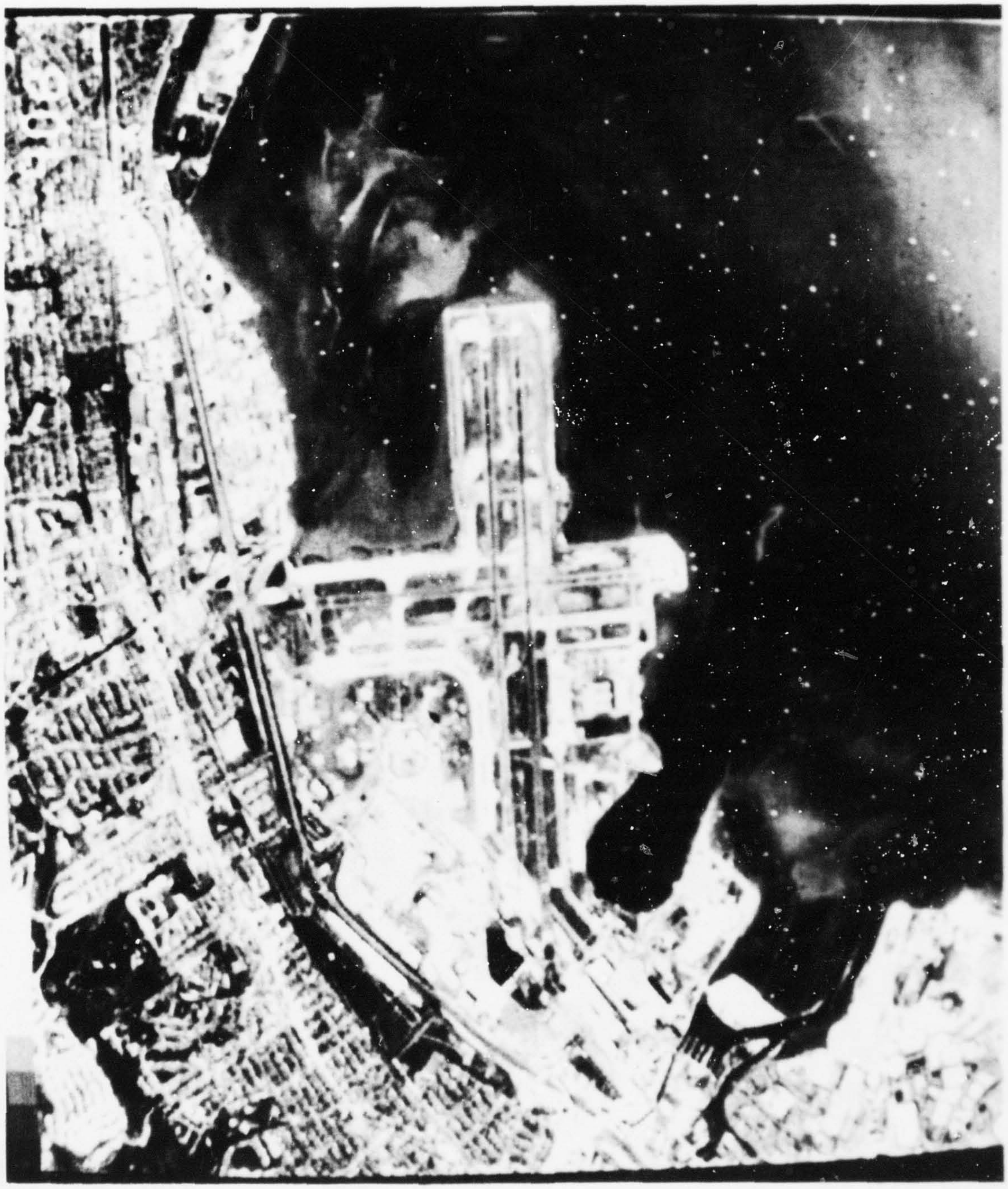


Fig. III-4: AP2 reconstructed from a PCM transmission over a noisy channel having an error rate of 10^{-3}. Photo number 406.

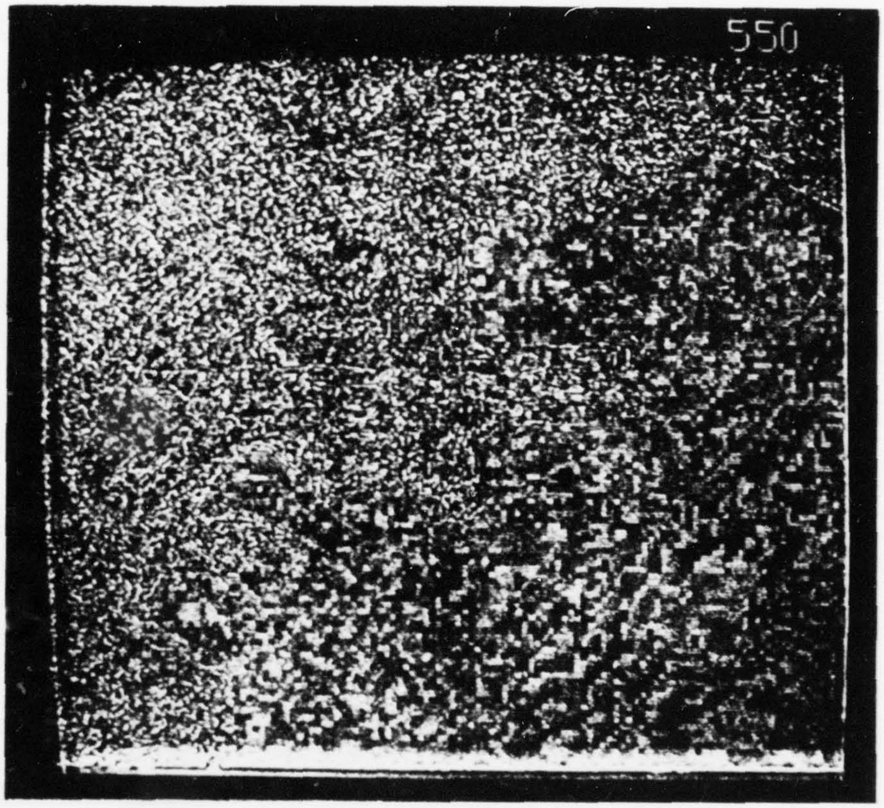


Fig. III-5: Difference picture yielded by block truncation as applied to AP2 at a compression of 1.5 bits/pixel. Photo number 550.

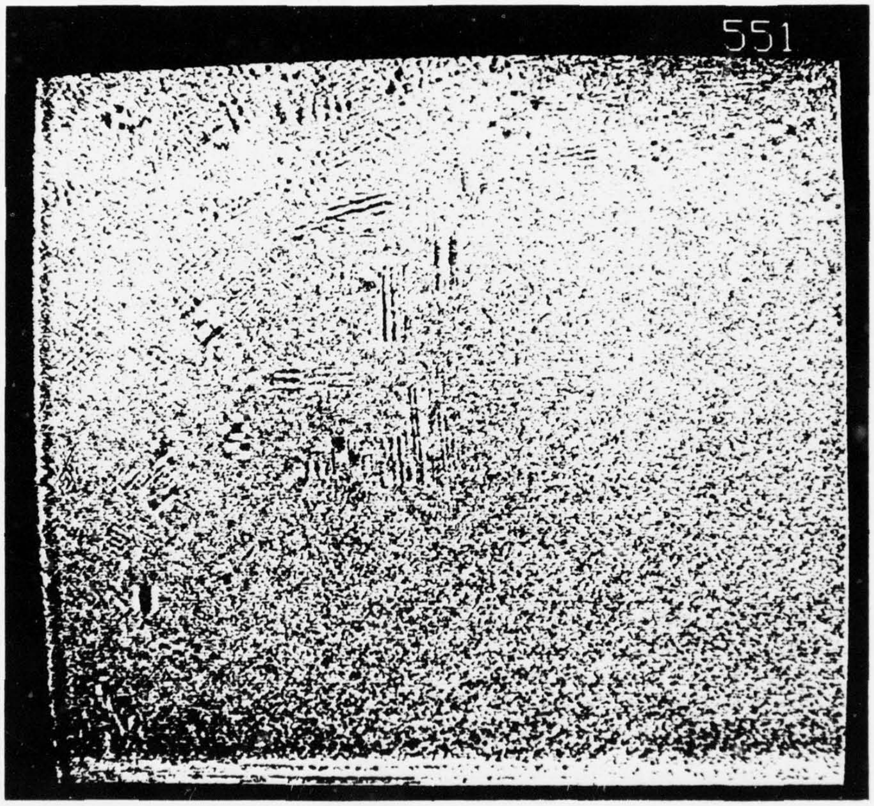


Fig. III-6: Difference picture yielded by Chen and Smith as applied to AP2 at a compression of 1.5 bits/pixel. Photo number 551.

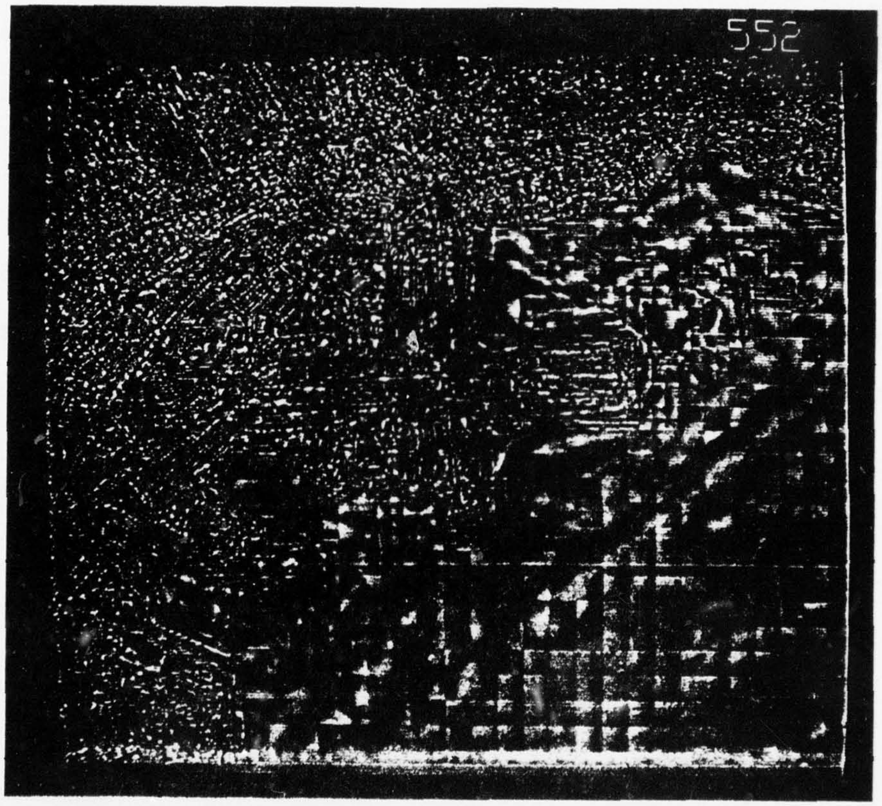


Fig. III-7: Difference picture yielded by MAPS as applied to AP2 at a compression of 1.5 bits/pixel. Photo number 552.

the case of the spatial techniques.

Of the twelve sets of photos ranked by the photo-analysts at Rome Research Corporation, three are shown here. The first set in Figs. III-8 to III-14, represent the application of seven coding techniques to AP2. The compression rate in each case was 1.5 bits/pixel. No channel errors were introduced. The second set of photos, Figs. III-15 to III-20, represent the same coding techniques and compression rate, as previously, but a channel error probability of 10^{-3} was introduced. This set does not include threshold coding because of synchronization problems discussed in Section II. The third set, Figs. III-21 to III-27, show all seven methods applied to SAM1 at a compression rate of 0.5 bits/pixel with no channel error.

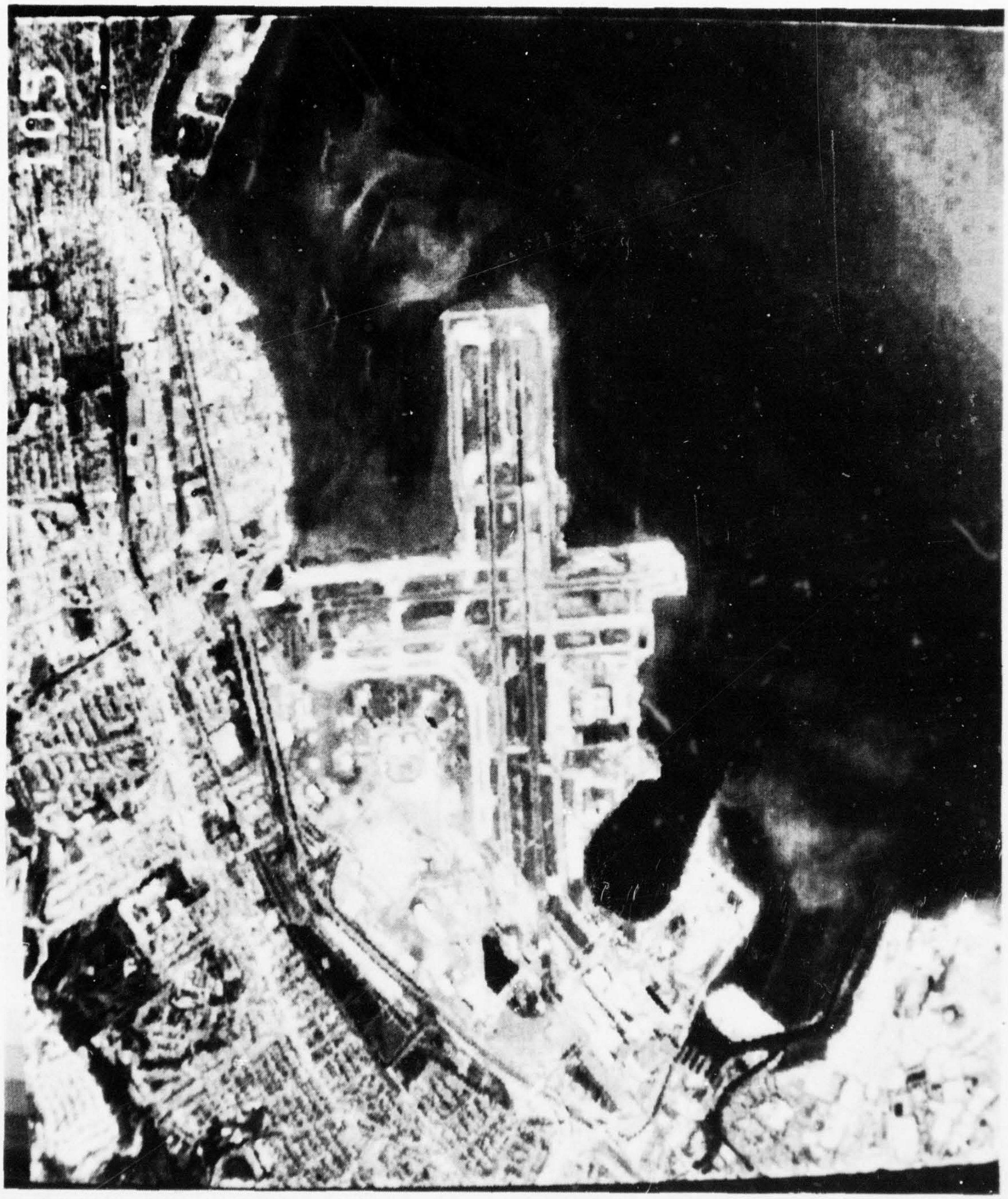


Fig. III-8: Reconstructed image using block truncation coding, AP2, 1.5 bits/pixel, 0 errors. Photo number 105.

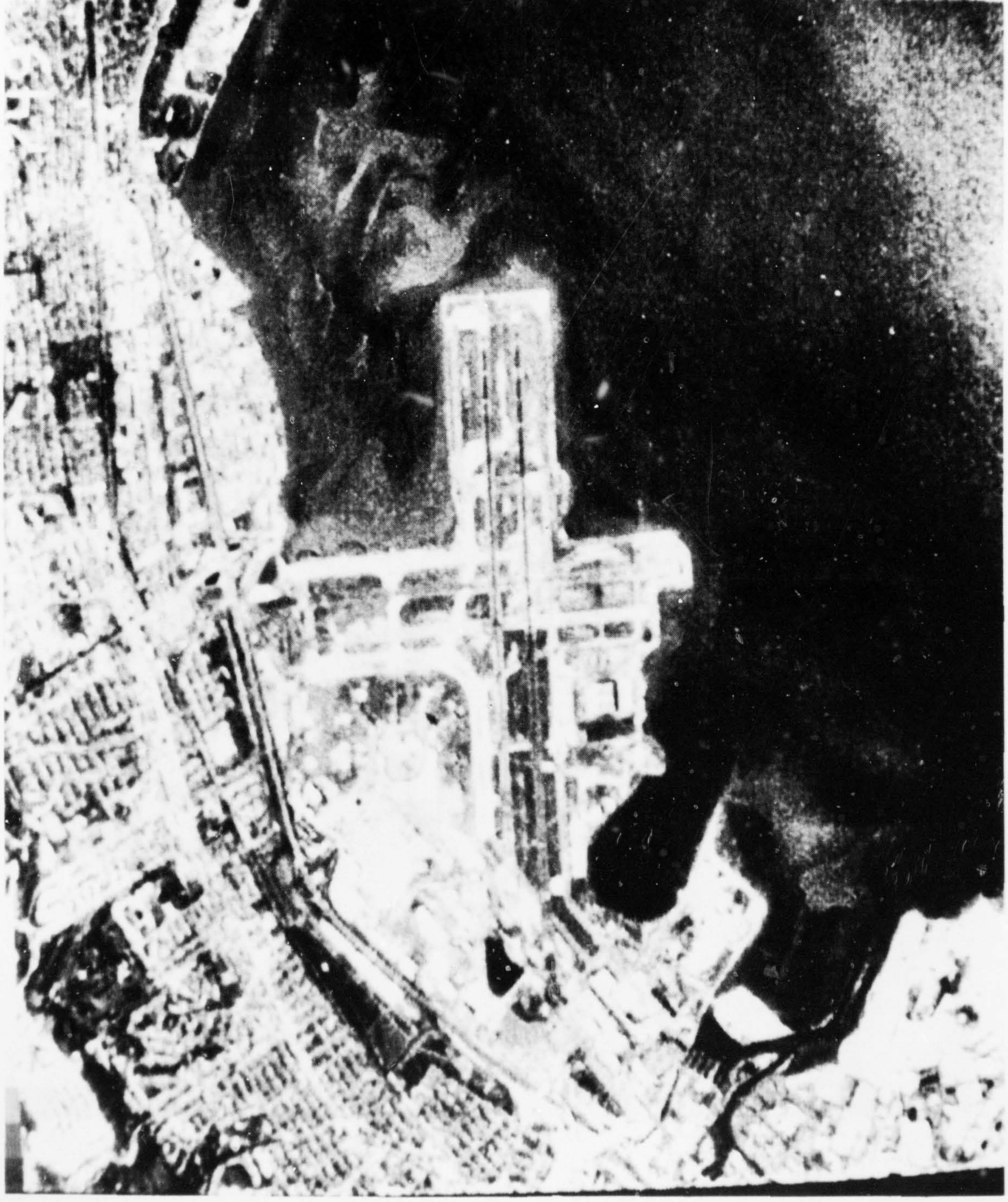


Fig. III-9: Reconstructed image using fixed zone coding, AP2, 1.5 bits/pixel, 0 errors. Photo number 749.

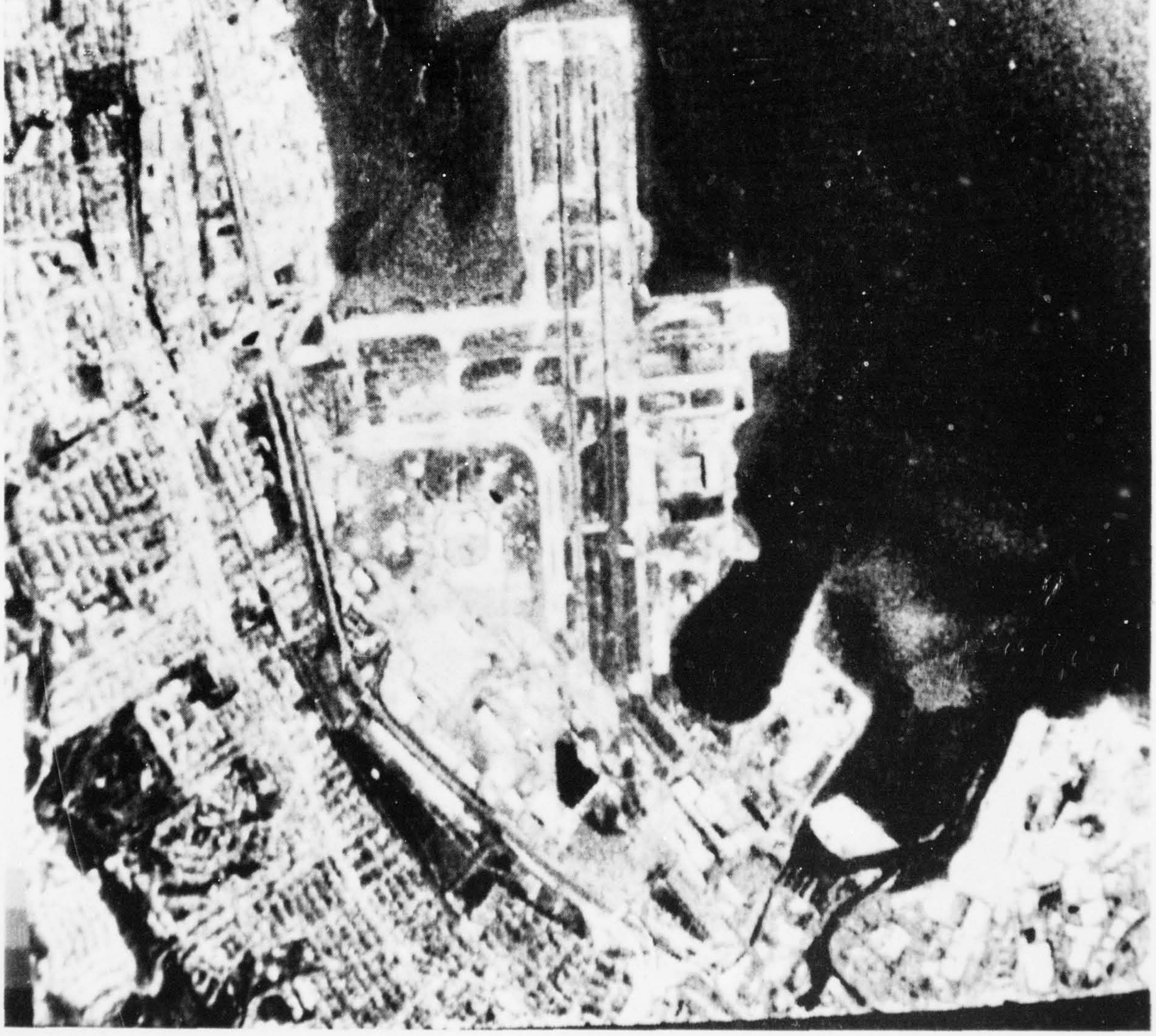


Fig. III-10: Reconstructed image using Chen and Smith coding, AP2, 1.5 bits/pixel, 0 errors. Photo number 720.

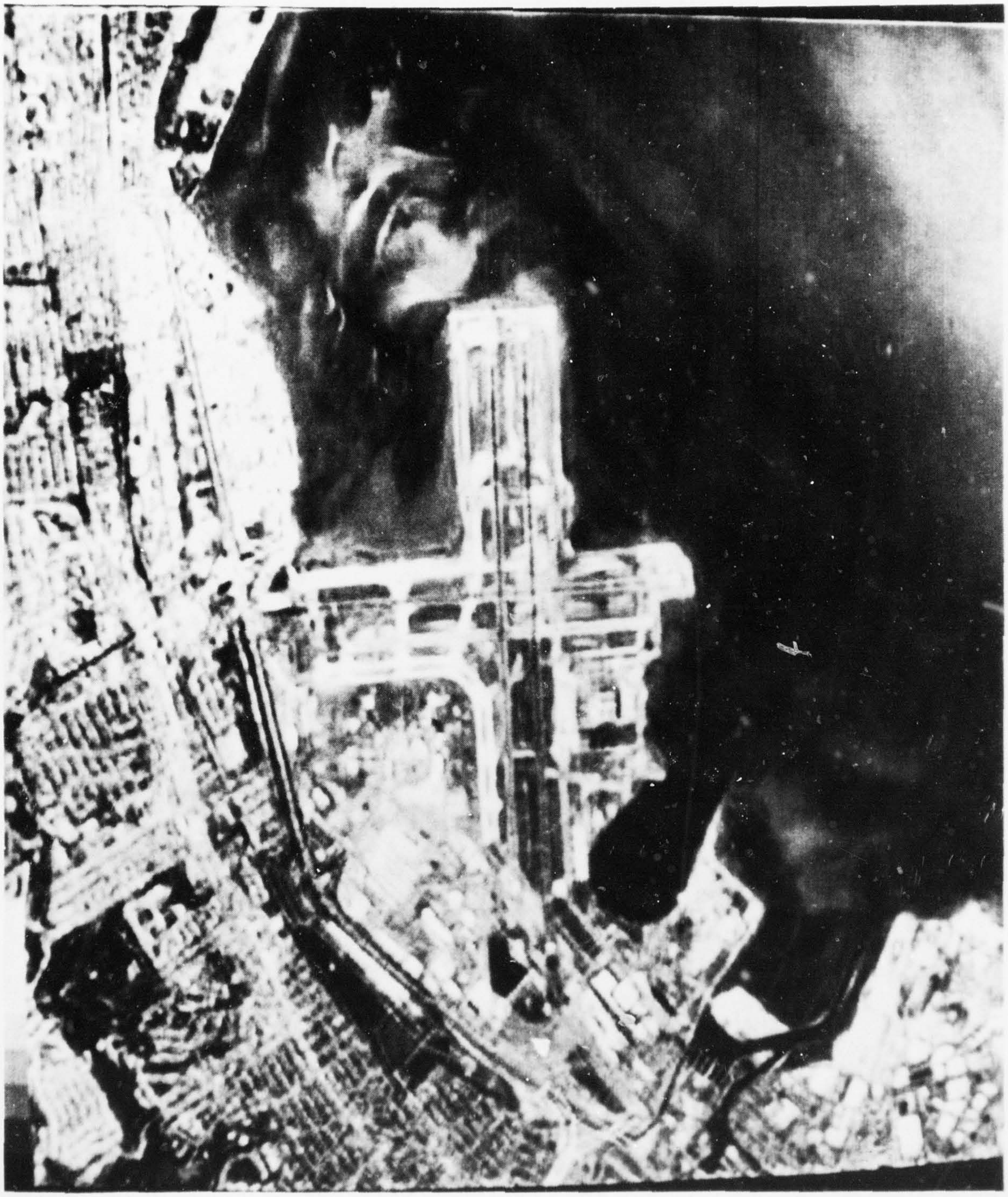


Fig. III-11: Reconstructed image using threshhold coding, AP2, 1.5 bits/pixel, 0 errors. Photo number 124.

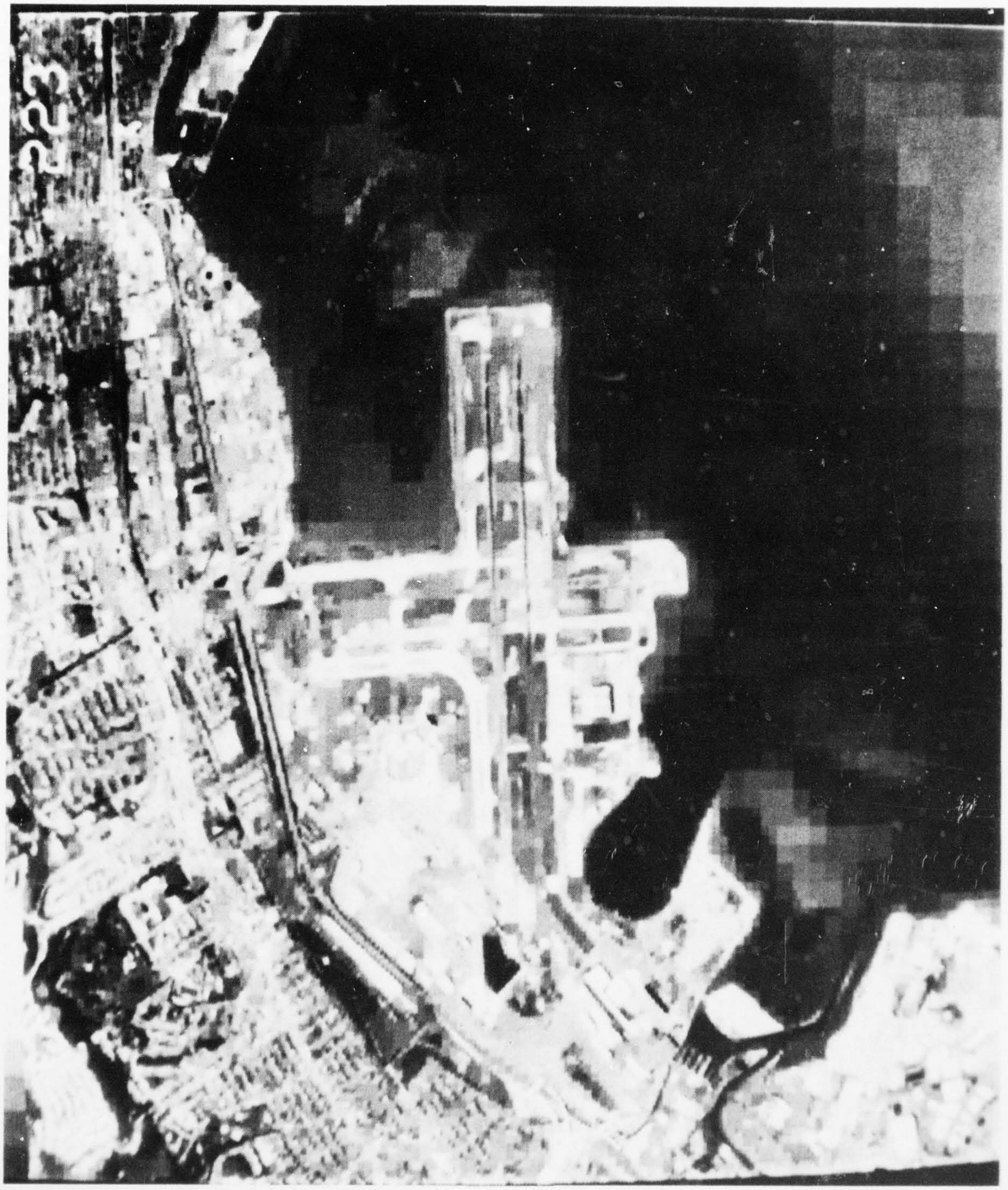


Fig. III-12: Reconstructed image using MAPS coding, AP2, 1.5 bits/pixel, 0 errors. Photo number 223.

Fig. III-13: Reconstructed image using variable zone coding, AP2, 1.5 bits/pixel, 0 errors. Photo number 750.

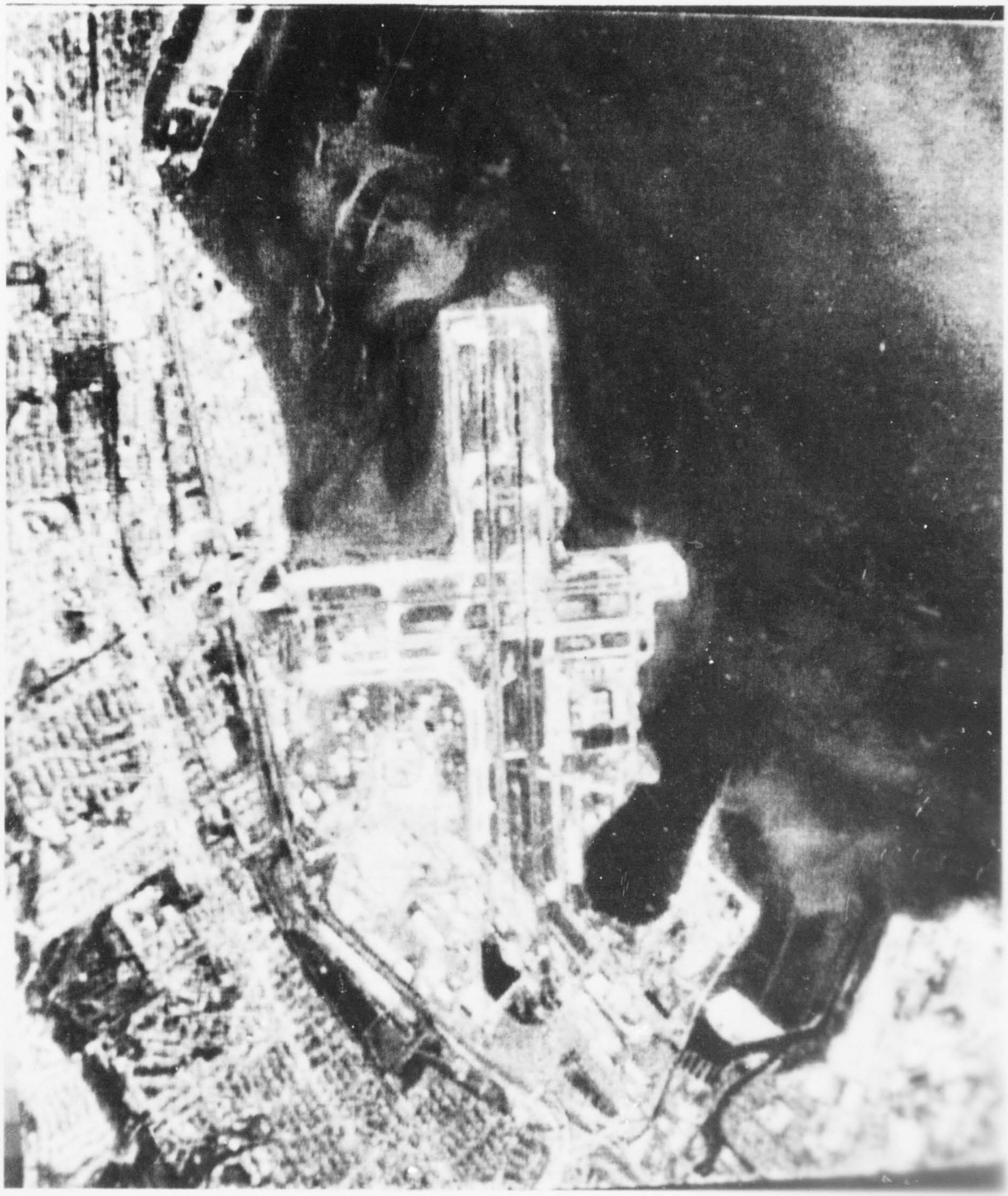
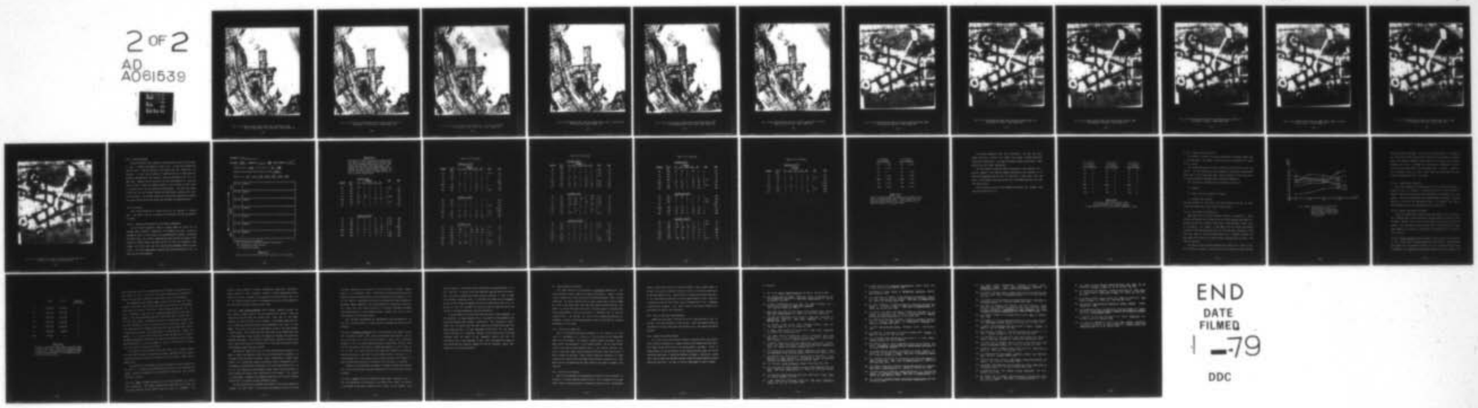


Fig. III-14: Reconstructed image using hybrid coding, AP2, 1.5 bits/pixel, 0 errors. Photo number 722.

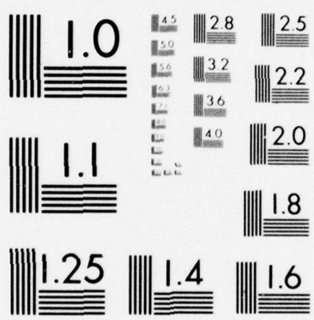
AD-A061 539 PURDUE UNIV LAFAYETTE IND SCHOOL OF ELECTRICAL ENGI--ETC F/G 9/4
CODING OF AERIAL RECONNAISSANCE IMAGES FOR TRANSMISSION OVER NO--ETC(U)
SEP 78 O R MITCHELL, S C BASS, E J DELP F30602-75-C-0082

UNCLASSIFIED RADC-TR-78-210 NL

2 OF 2
AD
A061539



END
DATE
FILMED
1 -79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

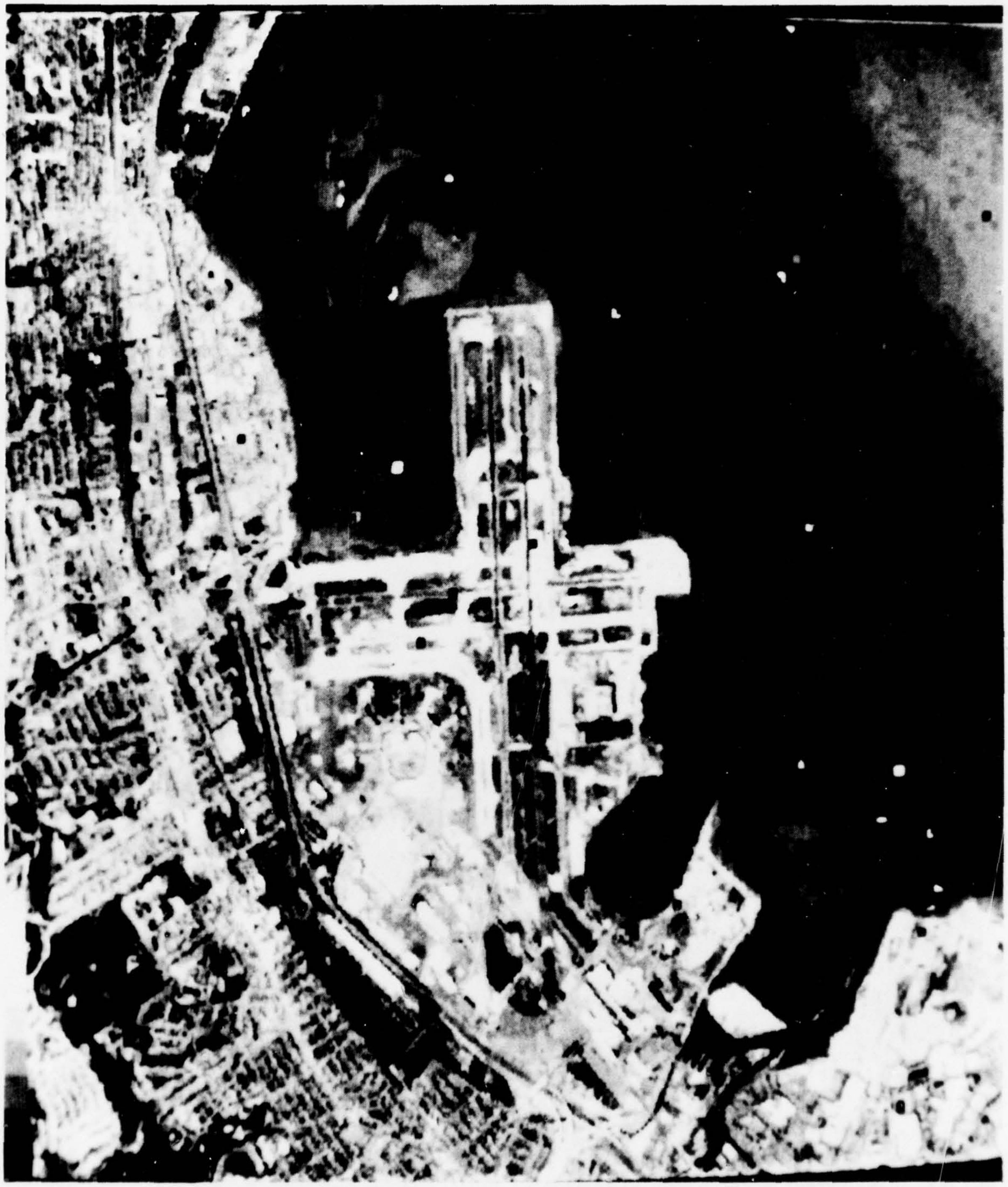


Fig. III-15: Reconstructed image using block truncation coding, AP2, 1.5 bits/pixel, 10^{-3} error rate. Photo number 112.

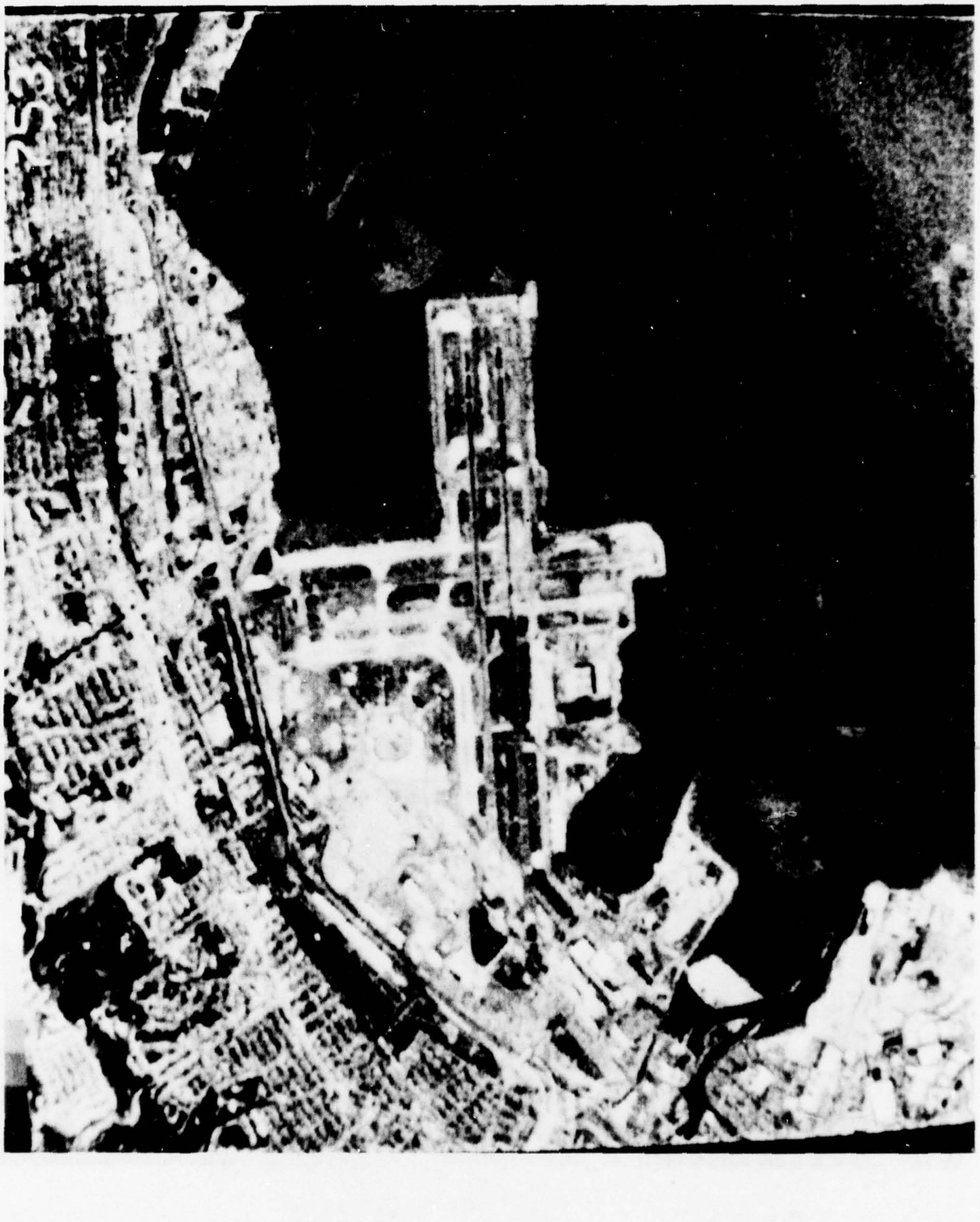


Fig. III-16: Reconstructed image using fixed zone coding, AP2, 1.5 bits/pixel, 10^{-3} error rate. Photo number 753.

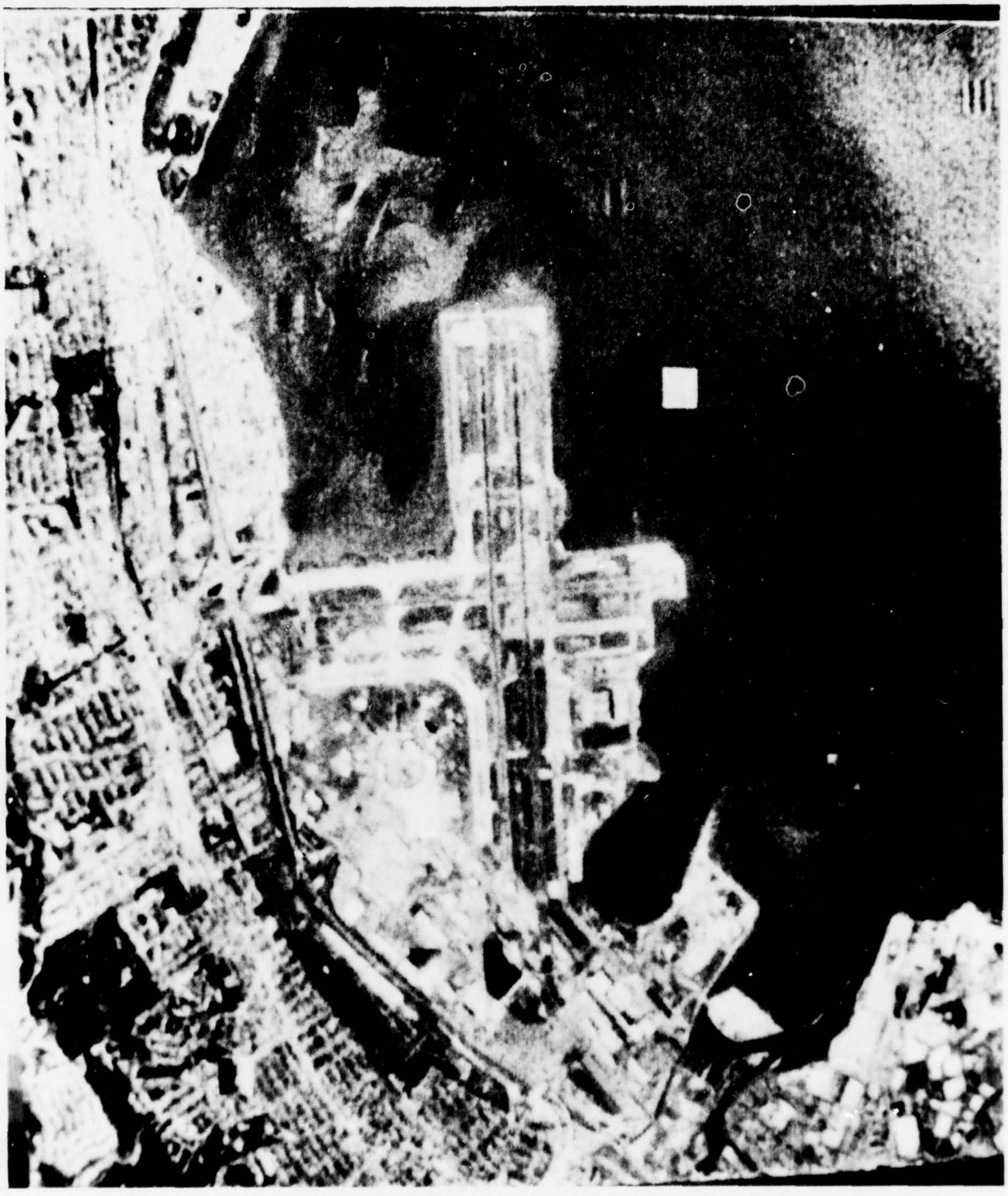


Fig. III-17: Reconstructed image using Chen and Smith coding, AP2, 1.5 bits/pixel, 10^{-3} error rate. Photo number 721.

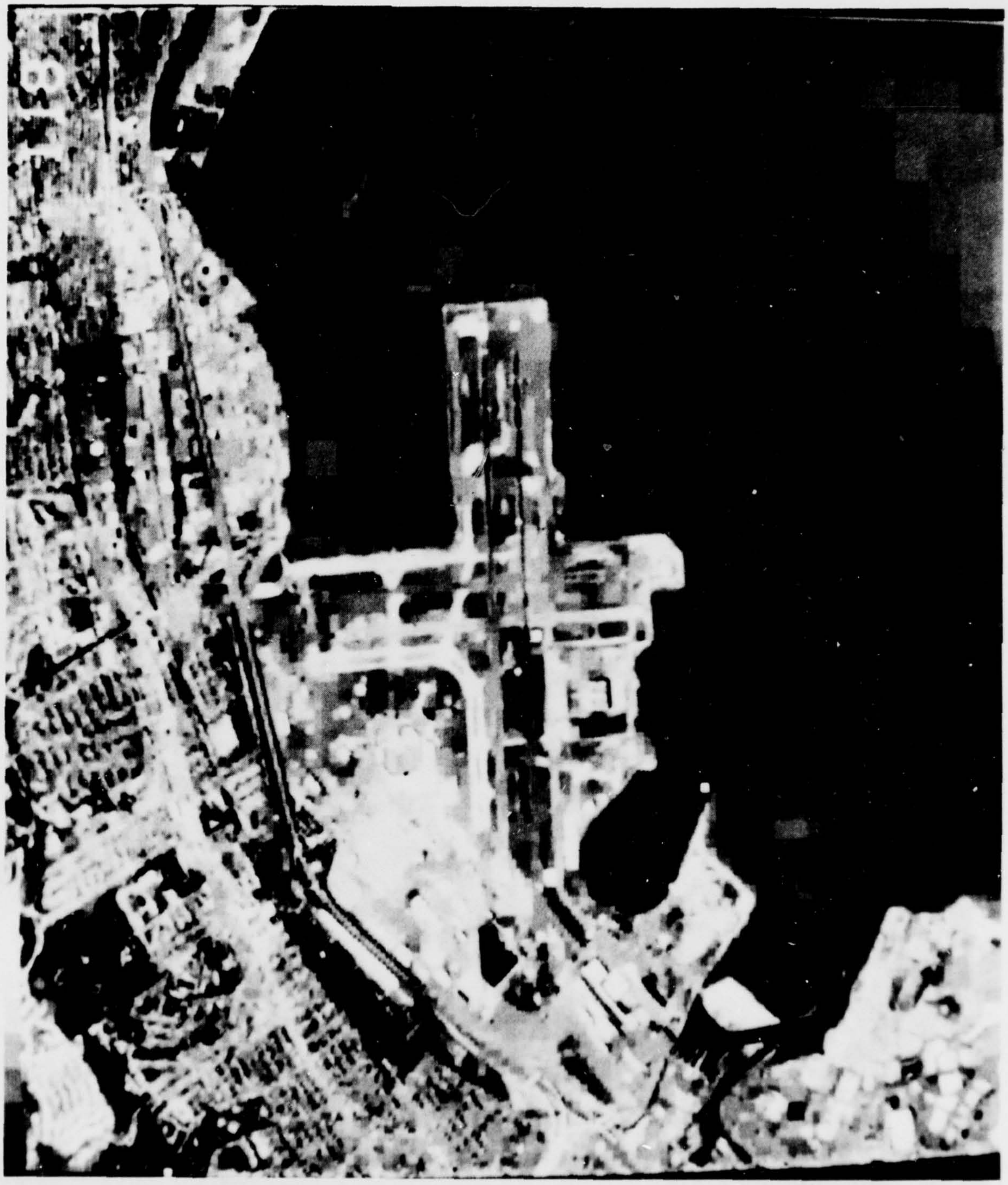


Fig. III-18: Reconstructed image using MAPS coding, AP2, 1.5 bits/pixel, 10^{-3} error rate. Photo number 118.

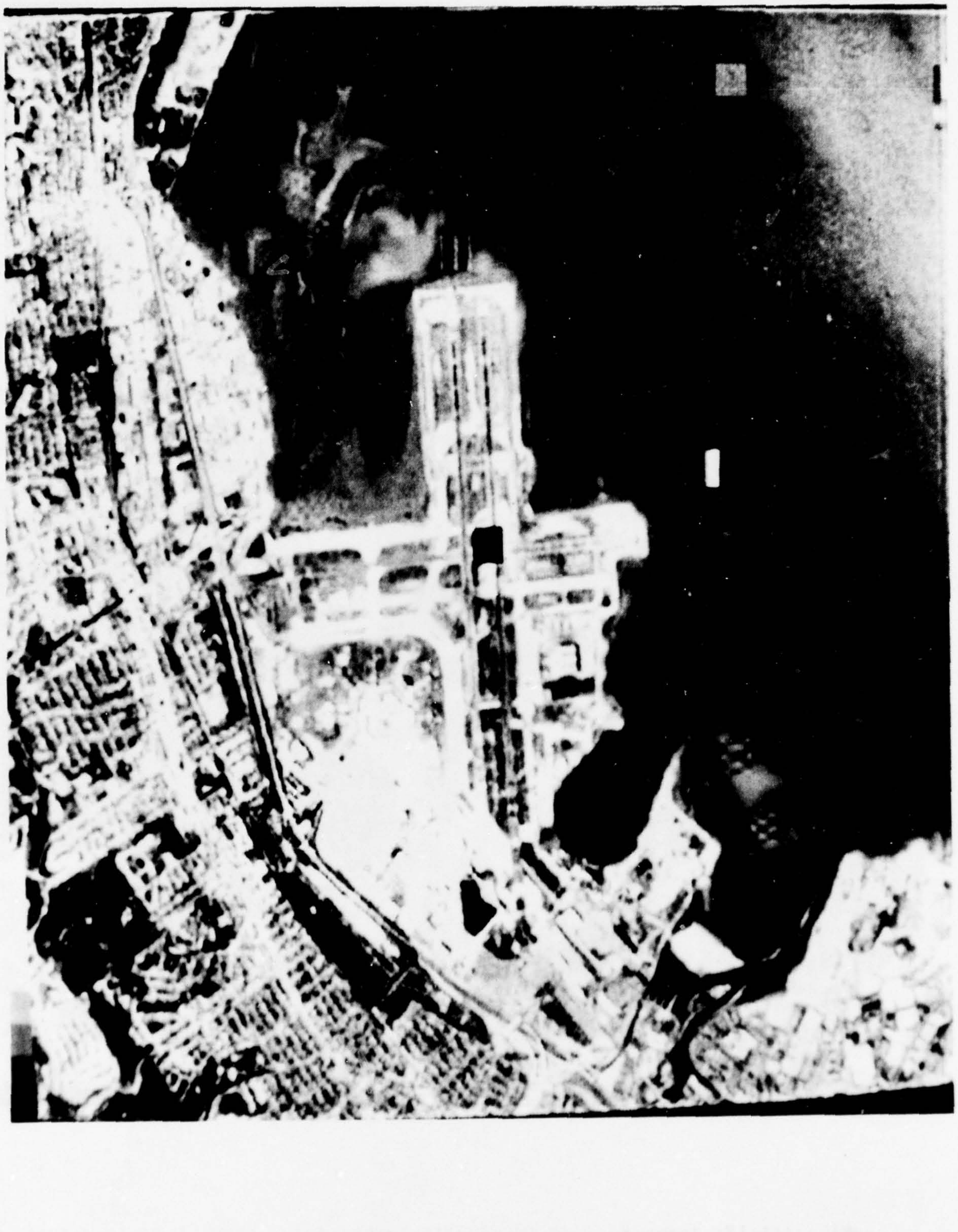


Fig. III-19: Reconstructed image using variable zone coding, AP2, 1.5 bits/pixel, 10^{-3} error rate. Photo number 754.

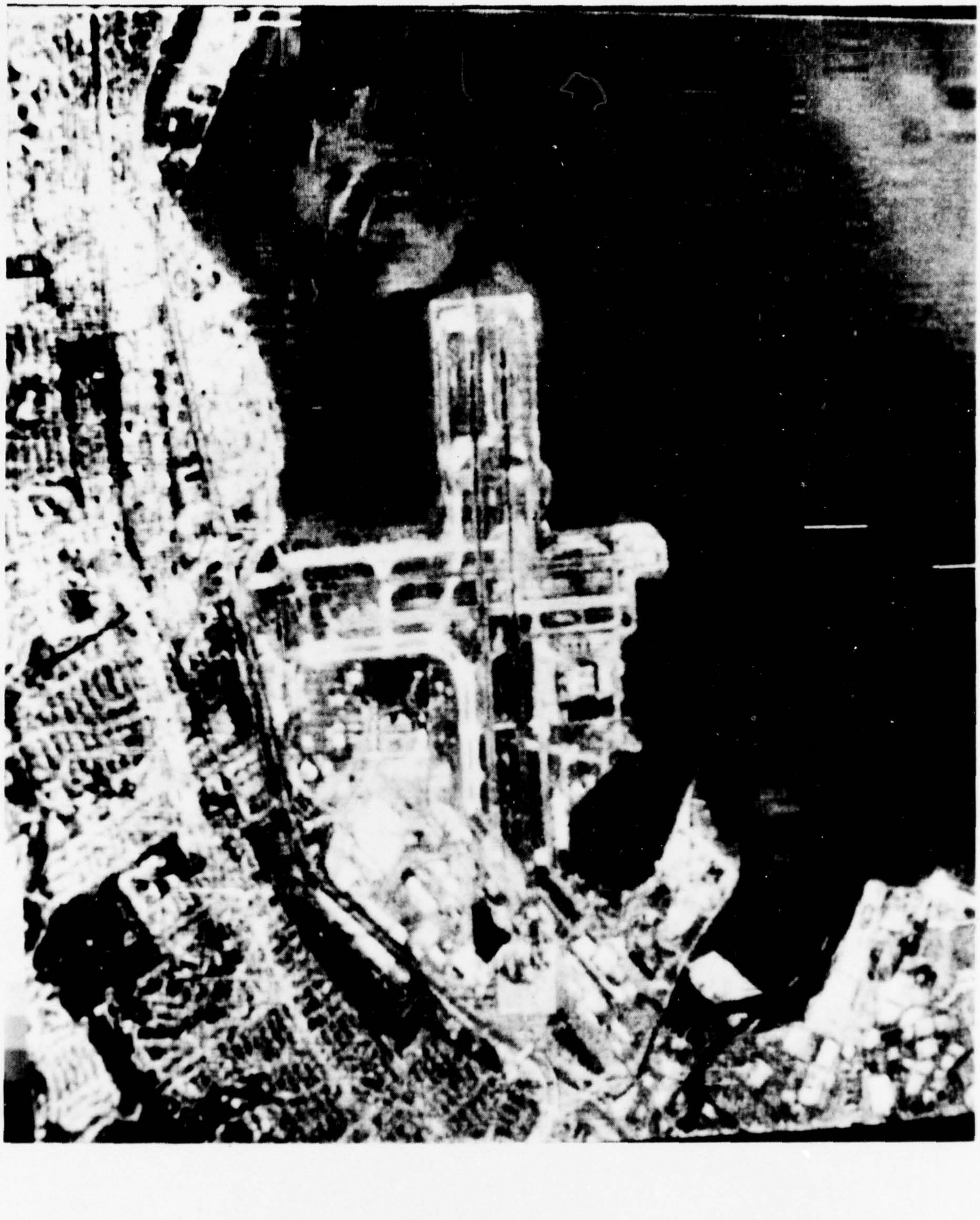


Fig. III-20: Reconstructed image using hybrid coding, AP2, 1.5 bits/pixel, 10^{-3} error rate. Photo number 724.

Fig. III-21: Reconstructed image using block truncation coding, SAMI, 0.5 bits/pixel, 0errors. Photo number 227.

Fig. III-22: Reconstructed image using fixed zone coding, SAMI, 0.5 bits/pixel, 0 errors. Photo number 729.

Fig. III-23: Reconstructed image using Chen and Smith coding, SAM1, 0.5 bits/pixel, 0 errors. Photo number 701.

Fig. III-24: Reconstructed image using threshhold coding, SAM1, 0.5 bits/pixel, 0 errors. Photo number 226.

Fig. III-25: Reconstructed image using MAPS coding, SAM1, 0.5 bits/ pixel, 0 errors. Photo number 101.

Fig. III-26: Reconstructed image using variable zone coding, SAM1, 0.5 bits/pixel, 0 errors. Photo number 730.

Fig. III-27: Reconstructed image using hybrid coding, SAM1, 0.5 bits/pixel, 0 errors. Photo number 714.

III.3. Raw Ranking Data

The photo analysts were supplied a ranking form for each of the 12 photo sets. A sample form appears in Table III-1. In Table III-2 appear the ranking results. Notice that each of the twelve sets was ranked by five analysts. In each case a ranking of 1 denoted the best (in that analyst's opinion) reproduction of the original. Space was provided on the form for additional comments such as excellent (EX) and unacceptable (X). The presence of either of these two comments appears in the raw data of Table III-2 as a superscript to the corresponding ranking number. The right two columns of each set list the "average ranking" and mean square reconstruction error for each method. The average ranking was calculated by dropping the highest and lowest ranking given each method and averaging the remaining three.

III.4. Discussion

The raw data presented in section III.3 can be analyzed in various ways. We would like now to present a few tentative conclusions based on this data.

III.4.1. Subjective Evaluation for Error-Free Transmission

Let the "quality potential" (QP) of a coding method be taken as its average photo analyst's ranking at a zero channel error rate. At the low compression rate (1.5 bits/pixel), the two-dimensional transform techniques performed best, the hybrid technique was next, and the two spatial techniques were ranked lowest (see Table III-3). Of the 2-D transform techniques, the fixed zone and the Chen and Smith methods exhibit the best Q.P.'s. The third place method, variable zone, performed about as well as these top two rated methods.

INTERPRETER INITIALS ____________

SET NUMBER A2 COMPRESSION 1.5 $\frac{\text{bits}}{\text{pixel}}$ ERROR PROBABILITY 10^{-4}

Original Picture 300 is 512 x 512 at 8 $\frac{\text{bits}}{\text{pixel}}$

Original Picture using PCM with same error rate 400

Photos in set 115, 127, 308, 725, 728, 755, 756

BEST	PIC. NO.	COMMENTS:
	PIC. NO.	COMMENTS:
	PIC. NO.	COMMENTS:
RANKING	PIC. NO.	COMMENTS:
	PIC. NO.	COMMENTS:
	PIC. NO.	COMMENTS:
WORST	PIC. NO.	COMMENTS:

Suggested abbreviations for comments:

EX = excellent (no significant difference from original)

↕ = two methods are about the same

X = unacceptable (useless)

TABLE III-1:

Sample rating form used by the photo analysts in this project.

TABLE III-2:

"Raw data" as taken from analysts rating forms. Also shown for each method are an average ranking and the computed mean-square error for that method when applied to the scene noted. The average rank was calculated after deleting the highest and lowest rating awarded a method. The MSE's include the effects of channel errors as well as coding inaccuracies.

A1(AP2-1.5-0)

Method	Photo No.	Analyst #1	#2	#3	#4	#5	AVG	MSE
BT	105	3^{EX}	6	2	6	4	4 1/3	94
FZ	749	5	4	3	2	5	4	82
CS	720	6	3	7	3	6	5	72
Th	124	1^{EX}	1^{EX}	6	5	2	2 2/3	75
Maps	223	7^{X}	7^{X}	4	7^{X}	7^{X}	7	89
VZ	750	2^{EX}	5	1	4	1	2 1/3	54
Hyb	722	4	2	5	1	3	3	152

A2(AP2-1.5-10^{-4})

Method	Photo No.	Analyst #1	#2	#3	#4	#5	AVG	MSE
BT	115	5	4	1	5	4	3 1/3	98
FZ	755	2	3	4	2	1	2 1/3	83
CS	728	3	2	2	3	2	2 1/3	72
Th	127	6	6	7	6	6^{X}	6	1062
Maps	308	7^{X}	7^{X}	6	7^{X}	7^{X}	7	93
VZ	756	1	1	3	1^{EX}	3	1 2/3	54
Hyb	725	4	5	5	4	5	4 2/3	158

Table III-2 continued

A3 (AP2-1.5-10^{-3})

Method	Photo No.	Analyst #1	#2	#3	#4	#5	AVG	MSE
BT	112	3	5^{X}	6	3	1	3 2/3	133
FZ	753	2	1	2	2	4	2	99
CS	721	1	2	3	1	5	2	111
Th	-	-	-	-	-	-	-	7038
Maps	118	6^{X}	6	1	6^{X}	3	5	108
VZ	754	4	3	4	4^{X}	6	4	95
Hyb	724	5	4	5	5^{X}	2	4 2/3	205

A4 (AP2-1.5-10^{-2})

Method	Photo No.	Analyst #1	#2	#3	#4	#5	AVG	MSE
BT	107	1	1	1	1	1	1	440
FZ	751	2	2	3	2	2	2	454
CS	727	3	4^{X}	4	3	4	3 2/3	477
Th	-	-	-	-	-	-	-	15783
Maps	301	5^{X}	3^{X}	2	5^{X}	5	4 1/3	1288
VZ	752	4	5^{X}	5	4^{X}	3	4 2/3	468
Hyb	723	6^{X}	6^{X}	6	6^{X}	6^{X}	6	781

B1 (SAM1-1.5-0)

Method	Photo No.	Analyst #1	#2	#3	#4	#5	AVG	MSE
BT	116	5	4	4	6	3	4 1/3	38
FZ	731	2^{EX}	5	2^{EX}	2^{EX}	1^{EX}	2	23
CS	702	1^{EX}	3	1^{EX}	1^{EX}	2^{EX}	1 1/3	23
Th	302	4	2^{EX}	5	4	6	4 1/3	27
Maps	103	7	7	3	7	7	7	45
VZ	732	6	6	7	5	4	5 2/3	23
Hyb	715	3^{EX}	1^{EX}	6	3	5	3 2/3	27

Table III-2 continued

Method	Photo No.	Analyst #1	#2	#3	#4	#5	AVG	MSE
		B2(SAM1-0.5-0)						
BT	227	4^{X}	3	4	3	3	3 1/3	95
FZ	729	2	1	3	1	2	1 2/3	49
CS	701	1	2	1	2	1	1 1/3	51
Th	226	5^{X}	5^{X}	5	4^{X}	5	5	74
Maps	101	3	4	2	5^{X}	4	3 2/3	144
VZ	730	7^{X}	6^{X}	7	6^{X}	6^{X}	6 1/3	87
Hyb	714	6^{X}	7^{X}	6	7^{X}	7^{X}	6 2/3	195
		C1(SAM2-1.5-0)						
BT	102	6	6	3	6	4^{EX}	5 1/3	56
FZ	741	4^{EX}	1^{EX}	1^{EX}	3	3^{EX}	2 1/3	17
CS	704	1^{EX}	3^{EX}	4	1^{EX}	6	2 2/3	14
Th	408	2^{EX}	4^{EX}	5	5	2^{EX}	3 2/3	26
Maps	114	7	7	7	7	7	7	53
VZ	742	5	2^{EX}	2	2	1^{EX}	2	18
Hyb	716	3^{EX}	5	6	4	5	4 2/3	27
		C2(SAM2-1.5-10^{-3})						
BT	221	4	5	4	6	3	4 1/3	94
FZ	745	5	4	2	5^{X}	2	3 2/3	77
CS	710	1	2	3	1	4	2	52
Th	-	-	-	-	-	-	-	6130
Maps	222	6	6	6	4	6	6	78
VZ	746	2	3	5	2	5	3 1/3	85
Hyb	718	3	1	1	3	1	1 2/3	68

Table III-2 continued

C3(SAM2-0.5-0)

Method	Photo No.	Analyst #1	#2	#3	#4	#5	AVG	MSE
BT	305	5^X	4^X	3	5^X	3	4	152
FZ	733	2	2	2	2	1	2	55
CS	703	1	1	1	1	2	1	56
Th	307	3	3	5	3^X	4^X	3 1/3	79
Maps	111	6^X	6^X	4	6^X	6^X	6	132
VZ	734	4	5^X	6	4^X	5^X	4 2/3	107
Hyb	726	7^X	7^X	7	7^X	7^X	7	250

C4(SAM2-0.5-10^{-3})

Method	Photo No.	Analyst #1	#2	#3	#4	#5	AVG	MSE
BT	405	3^X	3^X	3	3	3	3	186
FZ	737	2	1	2	1	2	1 2/3	109
CS	706	1	2	1	2	1	1 1/3	106
Th	-	-	-	-	-	-	-	4871
Maps	108	5^X	5^X	5	4^X	5^X	5	144
VZ	738	4^X	4^X	4	5^X	4^X	4	127
Hyb	712	6^X	6^X	6	6^X	6^X	6	279

C5(SAM2-1.5-10^{-4})

Method	Photo No.	Analyst #1	#2	#3	#4	#5	AVG	MSE
BT	110	5	5	3	5	4	4 2/3	61
FZ	747	2^{EX}	3	5	2	5	3 1/3	21
CS	708	1^{EX}	1^{EX}	1^{EX}	1^{EX}	1	1	14
Th	122	7	7^X	7	7^X	7^X	7	498
Maps	117	6	6	6	6	6	6	56
VZ	748	4	4	4	4	3	4	21
Hyb	719	3^{EX}	2^{EX}	2	3	2	2 1/3	34

Table III-2 continued

C6(SAM2-1.5-10^{-2})

Method	Photo No.	Analyst #1	#2	#3	#4	#5	AVG	MSE
BT	113	1	1	2	1	1	1	417
FZ	743	5	5^X	3	3	5^X	4 1/3	427
CS	709	2	2	4	2	2	2	331
Th	-	-	-	-	-	-	-	16254
Maps	303	6	6^X	6	6	4	6	2070
VZ	744	4	3	5	4	3	3 2/3	458
Hyb	717	3	4	1	5	6^X	4	781

3 Pictures 1.5 - 0 errors		2 Pictures 0.5 - 0 errors	
BT	4 2/3	BT	4 1/6
FZ	2 7/9	FZ	2
CS	3	CS	1 1/6
Th	3 5/9	Th	3 5/6
MAPS	5 8/9	MAPS	6 1/2
VZ	3 1/3	VZ	5 1/6
HYB	3 7/9	HYB	5 1/3

TABLE III-3:

Overall rating averages as computed at 0 channel errors and at two compression rates. These figures help indicate the quality potential of a coding method as compared with others in the same study.

At the high compression rate (0.5 bits/pixel), the Chen and Smith method definitely exhibits the highest QP, though straight zone holds respectable second place. The remaining methods ranked significantly below these two at this higher compression.

The only measure of absolute quality provided by the analysts was a written comment of "EX" meaning "appears equivalent to the original" or "X" meaning "unacceptable quality." At 1.5 bits/pixel, fixed zone, Chen and Smith, and threshhold received several EX's, while MAPS received X's on AP2 (see Table III-4).

At 0.5 bits/pixel no EX's were awarded, but hybrid and variable zone received a majority of X's.

No. of "EX's" 3 Photos 1.5 - 0 errors		No. of "X's" 3 Photos 1.5 - 0 errors		No. of "X's" 2 Photos 0.5 - 0 errors	
BT	2	BT	0	BT	4
FZ	8	FZ	0	FZ	0
CS	7	CS	0	CS	0
Th	6	Th	0	Th	5
MAPS	0	MAPS	4	MAPS	5
VZ	3	VZ	0	VZ	7
HYB	3	HYB	0	HYB	8

TABLE III-4:

The total number of "excellent" (EX) and "unacceptable" (X) comments applied to each method at 0 errors and at two compression rates.

III.4.2. Channel Error Sensitivity

The effects of channel errors were simulated by corrupting random bits in the coded data. The change in ranking versus error probability is shown in Fig. III-28.

The ranking improvement of block truncation at the high error rate was drastic. All five interpreters were unanimous in ranking this coding technique "best" at this high noise level. Block truncation is the clear choice of a coding method for operation under high error probabilty.

Fig. III-28 also indicates that if a channel error rate is

a) unknown,

b) known to be low to medium in value, or

c) strongly time varying,

then Chen and Smith or fixed zone is the better method choice due to their robust performance over a wide range of noise levels.

III.4.3. Mean-Square Error Analysis

Mean-square error has been a dominant method of measuring a coding method's performance One can observe the correlation between mean-square error and the analyst's ranking in Table III-2. Often the mean square error is misleading. For example, in set SAM1-1.5-0, the variable zone method resulted in the same mean square error as the fixed zone, although the fixed zone was ranked 2 and the variable zone 5 2/3. In general, however, the mean square error does give a crude ranking of methods which correlate with observer rankings.

The type of errors involved determine the visibility of these errors. The difference pictures in Figs. III-5 to III-7 show the errors resulting

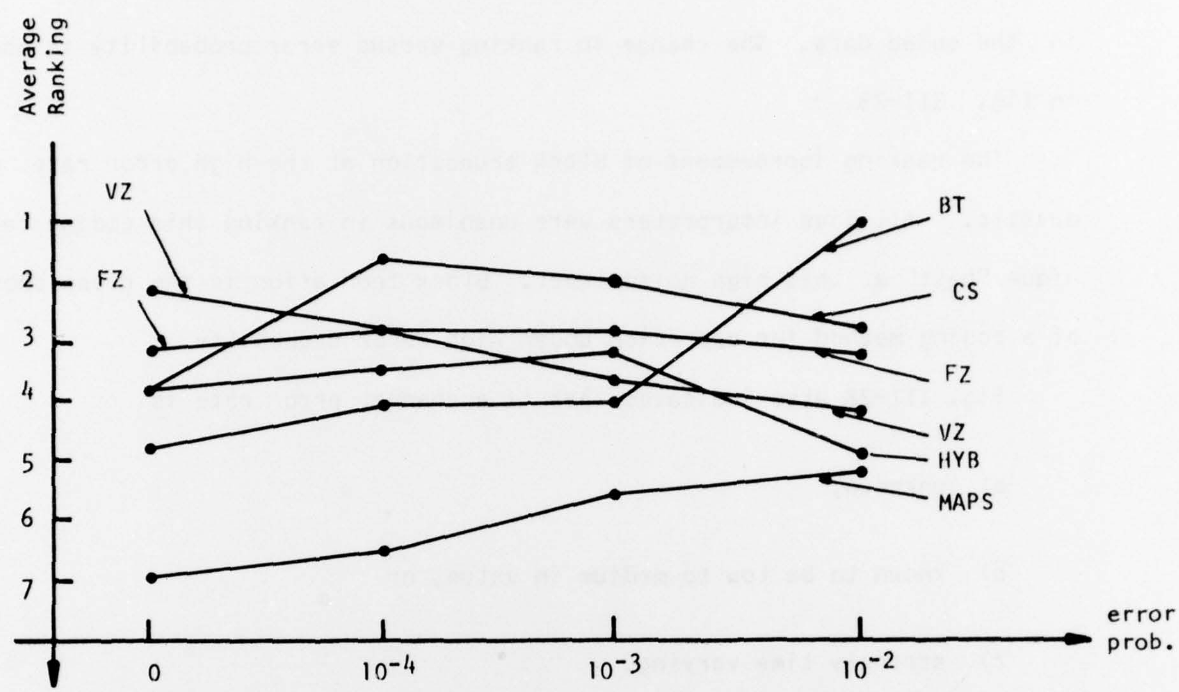


Fig. III-28: Average ranking variation with increasing error rate. Six coding methods. Average taken over two quite different scenes: "AP2" and "SAM2." Bit rate of 1.5 bits/pixel.

from three coding techniques. The top lines were omitted due to numbering on the pictures. Note that the sharp transition from the dark to the white band at the bottom of AIRPORT2 causes severe error in some of the techniques although this error is probably not significant to an observer. Also the errors are more correlated in some techniques than in others. Generally, the correlated errors are more visible than the uncorrelated ones, even though the mean-square error is the same.

III.4.4. Computational Complexity

Table III-5 summarizes the operations counts for coding and decoding a 512x512 picture as reported in Section II, and also lists the number of passes through the entire picture data required by the transmitter for each method. The counts exhibited by the transform domain techniques (including hybrid) all lie within the same general range. As expected, the spatial domain methods are the most computationally efficient of all.

III.4.5. Overall Performance Discussion

Except for computational complexity and high channel error performance, the two-dimensional transform methods at present seem to be the best choice of method. However, one can look at each coding result and ask, why did the observer rank that method lower than some other? In some cases, the answer seems apparent. Following is a discussion of each method and its potential in our opinion:

III.4.5.1. Block Truncation: Block truncation should preserve the resolution of the original while transform methods blur the original. Three problems were noted in the reconstructed results: (1) false contouring (see the water in AIRPORT2), (2) raggedness along straight edges, and (3) false edges. The

	# adds	# mults	# passes through data
FZ	5,233,000	3,149,000	2
CS	5,495,000	3,411,000	3
VZ	5,495,000	3,411,000	2
Th	5,483,000	3,149,000	2
Hyb	5,755,000	1,787,000	2
BT	1,078,000	578,000	1
MAPS	800,000	0	1

TABLE III-5:

Arithmetic operations count summary for each method tested in this study. These were originally given in section II. Also shown are the number of passes through the picture data (or its derivatives) required by each technique.

false contouring is due to the quantization of the mean of each block to too few levels (we only used 5 bits) and could be improved by differentially coding the block mean although this would affect the error performance.

The raggedness along straight edges is due to the fact that the binary quantizer must make a decision for each gray level that has an intermediate value relative to others in the block. Points along an edge often fall in this intermediate range and the quanitzer makes each one either dark or bright. This problem becomes especially severe when there is some noise in the gray level values. A possible solution is to preprocess the data (such as with a 3x3 median filter) to order the points in each small region. However, when dealing with imagery where each pixel may be significant (such as a runway one pixel wide), this preprocessing is inappropriate. Another approach is to vary the block size used (in much the same way as is done in the MAPS technique) so that difficult regions are handled as smaller blocks. Another approach is to vary the number of levels in each block; e.g., having two level blocks and three level blocks.

False edges can be caused by binary quantization of a sloping region. This might be minimized by transmitting a highly compressed (0.2 bits/pixel) transform image of the picture in addition to the currently coded spatial domain data.

The present error performance of block truncation is quite good as noted earlier. In addition, the computations required for this technique are modest. We feel the potential for this method is significant and warrants additional research.

III.4.5.2. MAPS: The MAPS technique is still under development by Control Data Corporation. The present implementation described in this report retains many important picture details (note the boats in AIRPORT2). This method is vastly superior in terms of computational complexity. The performance ranking of this technique improved at higher compressions and at higher error rates. However, the introduction of block artifacts and loss of low contrast details make this implementation currently non-competitive with the other techniques tested in this study.

III.4.5.3. Zonal Transform Methods: These methods presently produce the best overall coding results for low to medium error channels. These are fixed zone, Chen and Smith, and variable zone. Fixed zone is the simple standard by which the others can be compared. A noticeable granularity error in fixed zone and Chen and Smith can be observed in the water of AIRPORT2. This is due to the one bit quantization of many of the high frequency coefficients. This causes each such coefficient to be reconstructed with a fixed magnitude (only the sign can change) even though that coefficient might be quite small. This effect is significantly reduced in variable zone since these coefficients are uniformly set to zero when a block has minimal high frequency energy (see Fig. III-13).

One method to reduce this error in fixed zone and Chen and Smith is to apply a three level quantizer (1.6 bits) as the minimum bit assignment. In the case of Chen and Smith, different block categorization schemes might correct this problem and improve the method's overall performance as well. For example, classification of each block into four groups based on the ratio of low frequency to high frequency energy tends to group the water regions together much better than classifying on total energy. This results in considerably different bit assignment maps which set the high frequency coefficients in the water to more reasonable values.

The key decision in the variable zone method is the size of subzones to transmit for each block. In this study a percentage of total a.c. energy was used. However, this can result in the loss of small, isolated targets within a low bandwidth block. We believe an improved decision would be based on maximum reconstruction error as anticipted by the transmitter. Also the smaller transmitted zone sizes can automatically apply tapered bit assignment maps as opposed to the current practice of truncating an exciting full block map. This latter action will, of course, reintroduce the above mentioned granularity in low frequency blocks. However, the use of three level quantizers can be applied here.

Until the above improvements are implemented, fixed zone appears to be the best coding technique in terms of quality, computation, and error performance.

III.4.5.4. Threshhold Transform: Even in the presence of no channel errors, this technique yielded lower overall performance than expected. Part of this loss in quality is surely due to the need to apply error correction methods to the run length codes which specify coefficient location. Even with this protection, the method is judged unacceptable at error rates as low as 10^{-4}. Our more recent work has shown that a MAPS-like information ordering procedure is applicable to the threshold coding. The result is a coding method that is now usable at error rates as high as 10^{-3}. However for equal reconstruction quality, an additional 0.2 bits/pixel is required.

Presently this method does not appear to compete at these high compression rates with the zonal transform methods due to the coefficient location overhead.

III.4.5.5. Hybrid: The hybrid coding techniques raked reasonably well in the low compression (1.5 bits/pixel), low channel error cases. The failure of the method in the other situation can be traced to the channel error recovery problem. Since hybrid uses a differential coding technique, it is very sensitive to channel errors and therefore requires some error protection. We chose to update the picture every 64 lines with a line of PCM. This procedure showed two flaws: (1) the PCM line channel errors propagated as very visible streaks throughout the next 64 lines, and (2) the bits required to code the raw PCM line left very little to code the rest of the picture at the high compression rate (0.5 bits/pixel).

We are presently correcting these flaws by using a leaky integrator in the predictor to remove channel errors. The amount of leak depends on the correlation between each frequency coefficient. The correlation coefficients are calculated when the other statistics are done during the first pass through the data. The usual advantage of hybrid over the two dimensional transforms is in computing. However, since we had to go to a 32 long transform block for hybrid to get comparable results with the 2-D transforms, much of that advantage is lost. Until improvements are made, we feel hybrid does not presently compete with the 2-D transform coding techniques for our testing conditions.

IV. Future Research Directions

This report should not be construed as a conclusive comparison of coding techniques on aerial imagery for human photo analysis. However, we have tried to compare fairly many existing and developing methods under uniform conditions. As noted in Section III.4.5, each coding result could be judged as to why it was ranked lower than some other method and modifications suggested to improve that particular flaw. Of course, usually these modifications produce another round of results with a different set of peculiar characteristics. We believe the cycle is still producing worthwhile improvements. We feel the following research areas will provide the greatest improvements in this area.

IV.1. Criteria for Adaptivity

Many of the coding techniques are adaptive in the selection of zone size, bit assignments, or block size. The criterion for this adaptation needs to be investigated. For example, frequency domain techniques usually adapt based on measurements in the frequency domain. But are there better criteria that could be used--possibly even in the spatial domain; e.g., blocks containing edges should be handled differently than those blocks in which no edges are detected. These criteria should be sensitive to those reconstruction details that the human visual system is sensitive to. Thus, edges sould be reproduced but the exact height of the edge is probably less important.

IV.2. Channel Error Recovery

Many of the techniques are degraded by the need to insert overhead information to protect against channel errors. This is especially true when there is much of sensitive control information involved in the transmission (such as coefficient location in threshold coding). More research needs to be done in the area of receiver recovery of errors in the source code due to uncorrected channel errors. Large errors in reconstructed blocks can usually be determined by examining continuity among adjacent blocks. When this is detected, often the source error can be estimated and a more reasonable reconstruction produced. This would result in much better channel error performance for many of the techniques.

IV.3. Work on Individual Coding Methods

Suggestions were made in Section III.4.5 for improving many of the individual tecniques presented in that section. These improvements should be implemented, and where observable improvements occur, new comparisons should be made.

IV.4. Supervisory Coding Systems

It is very likely that different channel conditions (error rate and bit rate), quality and quantity of imagery desired, and hardware and time available impose varied constraints on the coding options available. Given an understanding of the performance of each coding technique under the various operating conditions, it should be possible to design a supervisory system which would choose and implement optimum coding strategies based on present operating conditions. The time is ripe for development of such a system.

V. References

1. W.K. Pratt, Digital Image Processing, John Wiley: New York, 1978.

2. G.B. Anderson and T.S. Huang, "Piecewise Fourier Transformation for Picture Bandwidth Compression," IEEE Trans. Communication, Vol. COM-20, pp. 488-491, June 1972.

3. N. Ahmen, T. Natarajan, and K.R. Rao, "On Image Processing and a Discrete Cosine Transform," IEEE Trans. Computers, Vol. C-23, pp. 90-93, Jan. 1974.

4. W.K. Pratt, W.H. Chen, and L.R. Welch, "Slant Transform Image Coding," IEEE Trans. Communication, Vol. COM-22, pp. 1075-1093, August 1974.

5. K.R. Rao, M.A. Narasimhan, and K. Revuluri, "Image Data Processing by Hadamard-Haar Transforms," IEEE Trans. Computers, vol. C-23, pp. 888-896, Sept. 1975.

6. H.C. Andrews, J. Kane, and W.K. Pratt, "Hadamard Transform Image Coding," Proc. IEEE, Vol. 57, pp. 58-68, Jan. 1969.

7. A. Habibi, "Hybrid Coding of Pictorial Data," IEEE Trans. Computers, vol. COM-22, pp. 614-624, May 1974.

8. J.B. O'Neal, "Predictive Quantizing System (Differential Pulse Code Modulation) for the Transmission of Television Signals," Bell System Tech. Journal, vol. 45, pp. 689-721, May-June 1966.

9. A. Habibi, "Comparison of Nth Order DPCM Encoder with Linear Transformation and Block Quantization Techniques," IEEE Trans. Communiction Tech., vol. COM-16, pp. 948-957, December 1971.

10. A.E. Labonte and C.J. McCallum, "Image Compression Techniques," Rome Air Development Center Technical Report, RADC-TR-77-405, A050679, Dec 1977.

11. O.R. Mitchell, E.J. Delp, and S.G. Carlton, "Block Truncation: A New Approach to Image Compression," Proceedings of the International Conference on Communications, Toronto, June 5-7, 1978.

12. T.N. Cornsweet, Visual Perception, Academic Press: New York, 1970.

13. W.-H. Chen and C.H. Smith, "Adaptive Coding of Monochrome and Color Images," IEEE Trans. Communication, vol. COM-25, pp. 1285-1292, November 1977.

14. L.D. Davisson, "Rate-Distortion Theory and Application," Proc. IEEE, vol. 60, pp. 800-828, July 1972.

15. J. Max, "Quantizing for Maximum Distortion," IRE Trans. Information Theory, vol. IT-6, pp. 7-12, March 1960.

16. N. Ahmed and K.R. Rao, Orthogonal Transforms for Digital Signal Processing, Springer-Verlag: Berlin, 1975.

17. G.G. Roussas, A First Course in Mathematical Statistics, Reading: Addison-Wesley, 1973.

18. J.F. Jarvis and C.S. Roberts, "A New Technique for Displaying Continuous Tone Images on a Bilevel Display," IEEE Trans. on Communications, Vol. COM-24, pp. 891-898, August 1976.

19. B. Lippel, "Comments on 'A New Technique for Displaying Continuous Tone Images on a Bilevel Display'," IEEE Trans. on Communications, Vol. COM-26, pp. 309-310, Feb. 1978.

20. J.F. Jarvis, C.N. Judice, W.H. Ninke, "A Survey of Techniques for the Display of Continuous Tone Pictures on Bilevel Displays," Computer Graphics and Image Processing, No. 2, pp. 13-40, 1976.

21. B. Lippel and M. Kurland, "The Effect of Dither on Luminance Quantization of Pictures," IEEE Trans. on Communication Technology, Vol. COM-29, pp. 879-888, Dec. 1971.

22. T. Berger, Rate Distortion Theory, Englewood Cliffs: Prentice-Hall, 1971.

23. Z.L. Budrikis, "Visual Fidelity Criterion and Modelling," Proceed. of IEEE, Vol. 60, pp. 771-779, July 1972.

24. T. G. Stockham, "Image Processing in the Context of a Visual Model," Proceed. of IEEE, Vol. 60, pp. 828-842.

25. O.R. Mitchell and E.J. Delp, "Image Noise Visibility and Fidelity Criteria," Proceedings of Electro-Optical Systems Design Conference, New York, Sept. 1976, pp. 209-217.

26. J.L. Munnos and D.J. Sakrison, "The Effects of a Visual Fidelity Criterion on the Encoding of Images," IEEE Trans. on Information Theory, Vol. IT-20, pp. 525-536, July 1974.

27. D.J. Sakrison, "On the Role of the Observer and a Distortion Measure in Image Transmission," IEEE Trans. on Communications, Vol. COM-25, pp. 1251-1267, Nov. 1977.

28. S.A. Kassam, "Quantization Based on the Mean-Absolute-Error Criterion," IEEE Trans. on Communications, Vol. COM-26, pp. 267-270, Feb. 1978.

29. J.M. Morris and V.D. Vandeline, "Robust Quantization of Discrete-Time Signals with Independent Samples," IEEE Trans. on Communications, Vol. COM-23, pp. 1897-1902, Dec. 1974.

30. R.G. Gallager, Information Theory and Reliable Communication, New York: John Wiley, 1968.

31. T.S. Huang, "Easily Implementable Suboptimum Runlength Codes," Conference Record, 1975 IEEE International Conference on Communications, Vol. I, June 16-18, 1975, pp. 7-8-7-11.

32. J.B. O'Neal, "Entropy Coding in Speech and Television Differential DPCM Systems," IEEE Trans. on Information Theory, Vol. IT-17, pp. 758-761, Nov. 1971.

33. L.G. Roberts, "Picture Coding Using Pseudo-Random Noise," IRE Trans. on Information Theory, Vol. IT-8, pp. 145-154, Feb. 1962.

34. E.J. Delp, R.L. Kashyap, O.R. Mitchell, and R.B. Abhyankar, "Image Modelling with a Seasonal Autoregressive Time Series with Applications to Data Compression," Proceedings of IEEE Conference on Image Processing and Pattern Recognition, May 31-June 2, 1978, Chicago.

35. L.E. Franks, "A Model for the Random Video Process," BSTJ, pp. 609-630, April 1966.

36. R.P. Roesser, "A Discrete State-Space Model for Linear Image Processing," IEEE Trans. on Auto. Control, Vol. AC-20, pp. 1-10, Feb. 1975.

37. A. Habibi, "Two-Dimensional Bayesian Estimate of Images," Proceed. of IEEE, Vol. 60, pp. 878-883, July 1972.

38. M.G. Strintzis, "Comments on 'Two-Dimensional Bayesian Estimate of Images'," Proceed. of IEEE, Vol. 64, pp. 1255-1257, Aug. 1976.

39. D.P. Panda and A.C. Kak, "Recursive Least Squares Smoothing of Noise in Images," IEEE Trans. on ASSP, Vol. ASSP-25, pp. 520-524, Dec. 1977.

40. J.W. Woods and C.H. Rademan, "Kalman Filtering in Two Dimensions," IEEE Trans. on Information Theory, Vol. IT-23, pp. 473-482, July 1977.

41. A. Habibi, "Survey of Adaptive Image Coding Techniques," IEEE Trans. on Comm., Vol. COM-25, pp. 1275-1284, Nov. 1977.

42. R.L. Kashyap and A.R. Rao, Dynamic Stochastic Models from Empirical Data. New York: Academic Press, 1976.

43. G.E.P. Box and G.M. Jenkins, Time Series Analysis-Forecasting and Control. San Francisco: Holden-Day, 1970.

44. J.W. Woods, "Two-Dimensional Discrete Markovian Fields," IEEE Trans. on Information Theory, Vol. IT-18, pp. 232-240, March 1972.

45. A. Rosenfeld and A.C. Kak, Digital Picture Processing. New York: Academic Press, 1976.

46. W.A. Gardner and L.E. Franks, "Characterization of Cyclostatioary Random Signal Processes," IEEE Trans. of Information Theory, Vol. IT-21, pp. 4-14, Jan. 1975.

47. N.J. Jayant, "Digital Coding of Speech Waveforms: PCM, DPCM, and DM Quantizers," Proceed. of IEEE, Vol. 62, pp. 611-632, May 1974.

48. D.K. Sharma and A.N. Netravali, "Design of Quantizers for DPCM Coding of Picture Signals," IEEE Trans. on Comm., Vol. COM-25, pp. 1267-1274, Nov. 1977.

49. W. Zschunke, "DPCM Picture Coding with Adaptive Prediction," IEEE Trans. on Comm., Vol. COM-25, pp. 1295-1302, Nov. 1977.

50. M.S. Bartlett, The Statistical Analysis of Spatial Pattern. Loneon: Chapman Hall, 1975.

51. B.H. McCormick and S.N. Jayaramanurthy, "Time Series Model for Texture Synthesis," International Journal of Computer and Information Sciences, Vol. 3, No. 4, pp. 329-343, 1974.

52. P. Whittle, "On Stationary Processes in the Plane," Biometrika, Vol. 41, Part 3 and 4, pp. 434-449, 1954.

53. C.K. Chow, B.L. Deekshatulu, and L.S. Luh, "Some Computer Experiments in Picture Processing for Data Compression," Computer Graphics and Image Processing, No. 3, pp. 203-214, 1974.